PAPERS IN COMPUTATIONAL LEXICOGRAPHY COMPLEX '92

Edited by
Ferenc Kiefer, Gábor Kiss and Júlia Pajzs

LINGUISTICS INSTITUTE
HUNGARIAN ACADEMY OF SCIENCES, BUDAPEST

PAPERS IN COMPUTATIONAL LEXICOGRAPHY
COMPLEX '92

PAPERS
IN COMPUTATIONAL LEXICOGRAPHY
COMPLEX '92

Edited by
Ferenc Kiefer, Gábor Kiss and Júlia Pajzs

LINGUISTICS INSTITUTE
HUNGARIAN ACADEMY OF SCIENCES, BUDAPEST
1992

Proceedings of the 2nd International Conference on
Computational Lexicography, COMPLEX '92
Budapest, Hungary

All correspondence should be sent to

Linguistics Institute, Hungarian Academy of Sciences
Department of Lexicography and Lexicology
Budapest P.O. Box 19
Hungary 1250

Cover design by Gábor Kiss

ISBN 963 8461 67 5

Contents

Preface

The present collection contains papers to be presented at the 2nd International Conference on Computational Lexicography (COMPLEX '92) to be held in Budapest, October 4 through 8, 1992, and organized jointly by the Research Institute for Linguistics of the Hungarian Academy of Sciences and Laboratoire Automatique Documentaire et Linguistique - Université Paris 7. The aim of the conference is to bring together researchers who are experts in both computational linguistics and lexicography. One of the central issues addressed in this collection concerns the machine-aided compilation of dictionaries on the basis of text corpora. The computer-aided process of producing lexical descriptions in most often based on domain-specific corpora. Most lexicographic work reported on in the papers is directed toward the compilation of monolingual dictionaries, though the preoccupation with the problems of bilingual dictionaries seem gaining ground. Thus, in computational lexicography one of the most intriguing questions is the use of bilingual reference corpora for translation. From among the more recent topics of research mention should also be made of the analysis of compounds and idioms, of the recognition of complex patterns (verb with their complements) and of the acquisition of semantic information. On the other hand, quite a few of the more traditional topics of computational lexicography are still not exhausted as testified by a number of papers. To these belong automatic lemmatization, automatic lexical search in various types of text corpora, morphological analysis and parsing techniques.

By and large, the papers in this collection provide an adequate picture of present-day work in computational lexicography and represent work of high standard.

Ferenc Kiefer

Tools for computer-aided corpus lexicogrphy: the Hector Project

B.T. SUE ATKINS

Abstract

This paper describes the lexicographical work being done, and the computer tools being designed and used, in a 2-year collaborative project undertaken by a large computer systems research centre and a major dictionary publisher. The lexicographical objective is to compile dictionary-database entries and at the same time manually sense-tag part of the electronic corpus used as evidence for the lexicography. The computational objective is to build customized tools to facilitate the lexicography. The focus in this account is on the lexicographical work. The steps in the process of compilation and manual sense-tagging are enumerated, and the computer tools that make this process easier and faster are described and illustrated.

0 Introduction

In this paper[1], I shall record the computational tools designed and built by members of the staff Digital Equipment Corporation in order to facilitate lexicography carried out by members of the staff of Oxford University Press [OUP]. After introducing this joint project (Section 1), I shall give a brief account of the resources available to the lexicographers (Section 2); next I shall describe the tools themselves (Section 3); and then I shall go through in some detail the whole process of compiling a lexical entry (Section 4), showing at every stage how the computer tools function, and illustrating each step with a screen dump taken in the course of the lexicographical work.

1. THE HECTOR PROJECT

The Hector project is the name given to a collaborative venture into computer-aided lexicography undertaken by the US computer company Digital Equipment Corporation and the British publishing house Oxford University Press. It is located at Digital's Systems Research Center in Palo Alto, CA, and administered by Digital. Its purpose is twofold: to compile traditional dictionary-type lexical entries using corpus evidence, and to sense-tag the corpus lines that are being scanned in the process. It was begun in January 1991 and will end in March 1993; four members of Digital's staff work full time on this project; several OUP lexicographers contributed to the design phase, and four are working in Palo Alto for a year; a number of others from both companies make contributions as required. I shall describe the project entirely from a lexicographer's point of view (for a computational view, see Glassman et al (forthcoming)), and to try to show how the computer tools make the work faster, more thorough, more consistent, and infinitely more fun.

The lexical entries constitute at present a source database rather than a dictionary text proper. Our purpose during this phase of compilation is to analyse the way words are used, and to record as much information as possible, which in effect means as much information as we can manage within our own time constraints.

The manual sense-tagging of the corpus occurrences of each compiled word is not a task that has (as far as I know) been undertaken on this scale before, although lexical research has skirted round this topic for some years (see Lesk (1986), Atkins (1987), Black (1988)). In

1 This is an account of a team project, managed by Mary-Claire van Leunen (for Digital Equipment Corporation) and Patrick Hanks (for Oxford University Press). The Digital team consisted of Lucille Glassman, Cynthia Hibbard, James R. Meehan, and Loretta Guarino Reid (DEC Systems Research Center, Palo Alto, CA). The principal Oxford lexicographers were Rosamund Moon and William R. Trumble, assisted at different stages by Helen Liebeck, Peter Gilliver, and Katherine Barber. My role was that of lexicographical adviser to the project, concentrating on the software specifications and including some practical lexicography. My thanks go to Mary-Claire van Leunen and Patrick Hanks for their comments on the first version of this paper.

the initial stages of compiling an entry it is easy to set conditions that will produce groups of occurrences to be given the same sense tag in a macro command, but as the manual tagging wears on this becomes less and less possible, and it has to be said that in the case of the words being tagged during this project (on average, those which occur between 500 and 1,000 times in the corpus) the last 50% or more of the manual sense-tagging is tedious and slow.

2. LEXICOGRAPHICAL RESOURCES

The resources available to the lexicographers are of various types: evidence of the language in use (see 2.1), bibliographical details of the corpus texts from which the evidence is drawn (see 2.2), reference works about the language (see 2.3), and the result of linguistic analysis of the English verb system (see2.4).

2.1 LINGUISTIC DATA

The main body of evidence is to be found in the electronic corpus; this is supplemented by a file of citations collected as part of OUP's continuing reading programme.

2.1.1 THE ELECTRONIC TEXT CORPUS

The bulk of the evidence to be analysed takes the form of an electronic corpus of current British English, containing approximately 17.3 million words principally of written texts but also some transcribed spoken language[2]. The corpus contents were preprocessed for the lexicographers: cleaned up (punctuation checked and tagged, duplicate texts and typographical errors removed, etc.); wordclass-tagged, using a tagset of 623 tags devised for the Lund-Oslo-Bergen corpus (Johansson & Hofland 1989), grouped into 14 clusters; parsed, using the Houghton-Mifflin parser, which includes a second tagging process; and wordform occurrences and collocational frequencies computed.

2.1.2 THE CITATIONS FILE

OUP maintains a reading programme in Oxford and (for American and Canadian texts) in New Jersey; the citations collected in the course of this programme are keyed and used by Oxford exicographers recording new words and new usages of existing Words; this continuously growing body of text is available on line to the lexicographers of the Hector Project.

2 This corpus was drawn from the Oxford Corpus, built along the lines described in Clear (in press); we are grateful to Jeremy Clear, Corpus Manager, OUP, for his help in building it.

2.2 CORPUS CATALOGUE

Bibliographical details of every text in the corpus may be called up by the lexicographers. A typical entry (with end tags removed) reads:

<code>	NiceWk	
<title>	Nice Work	
<comment>	novel. Booker shortlist	
<date>	1988	
<authper>	David Lodge	
<age>	40-50	
<authmode>	single	(= only one author)
<sex>	male	
<nationality>	UK	
<domicile>	UK	
<compos>	single	(= only one typesetter)
<publisher>	Secker and Warburg	
<place>	London, UK	
<genre>	written; published; books; fiction	
<samplen>	109,109	(= number of words in sample)

2.3 OXFORD REFERENCE WORKS[3]

The following works are available on line for consultation by the lexicographers:

The Concise Oxford Dictionary [COD8], 8th edition, Oxford University Press, 1990.

The Pocket Oxford Dictionary, 8th edition, Oxford University Press, 1992.

The Oxford Dictionary of Quotations [ODQ], 3rd edition, Oxford University Press, 1980.

The Oxford Thesaurus, compiled by L. Urdang, Oxford University Press, 1992.

The Shorter Oxford English Dictionary [SOED2], (text of forthcoming new edition), Oxford University Press.

The attributed citations in the SOED2 and ODQ, together with those in the Citations File (see 2.1.2), usefully complement the electronic corpus.

3 We are grateful to James Howes, Head of Reference Computing, Oxford University Press, for help with the transfer to Palo Alto of dictionary data and for the OUP "Sid" dictionary browsing software which contributes to the Argus Reference Tools and the Ajax Dictionary Editor.

2.4 VERB CHECKLIST

This is adapted from Levin (forthcoming), and consists of a listing of verb patterns and alternations; it is available on line, and serves as an aide-mémoire for lexicographers who wish to check that the corpus and citations sources offer adequate information about the constructions associated with the principal English verbs. When a lexicographer asks about a verb, the program offers one or more of the verb lists (see 4.1.4 for an example) where patterns of usage, including transitivity alternations, are stated; each pattern is exemplified, using one of the verbs to which that pattern applies, in such a way that the lexicographer can see the potential of the verb that is the subject of the query.

3. THE COMPUTATIONAL TOOLS[4]

In this paper, I wish to concentrate on the specifically lexicographical functions of the Hector tools. There are a number of more general functions which I shall not describe idiot-proofing (for instance, they don't allow lexicographers to destroy data without being aware of it); periodic automatic saving; monitoring and recording every command that is input, together with its consequences; flexible windowing for almost all of the menu and dialogue boxes, and so on.

The lexicographers' machines are Digital 5100 workstations. The lexicographical process, including corpus searching and sense-tagging, is too complex to be carried out on a single monitor, even one with many windows (the windows are organized by the Motif Window Manager). The prototype configuration had six monitors, and these were all used and useful, but the hardware eventually decided on meant that we finished up with three (see Figure 1). This was a relief to those whose eyesight could not cope with six screens all in different focus.

The computational tools designed to assist the lexicography fall into three main groups:

(1) those which access reference material, used in the left-hand monitor ("Atlas");

(2) those which display structured corpus texts, and perform operations on these, in the centre monitor ("Argus");

(3) those which receive, structure, record and display lexical entries and amendments to these, in the right-hand monitor ("Ajax").

4 This account owes much to the excellent documentation provided by our Digital colleagues.

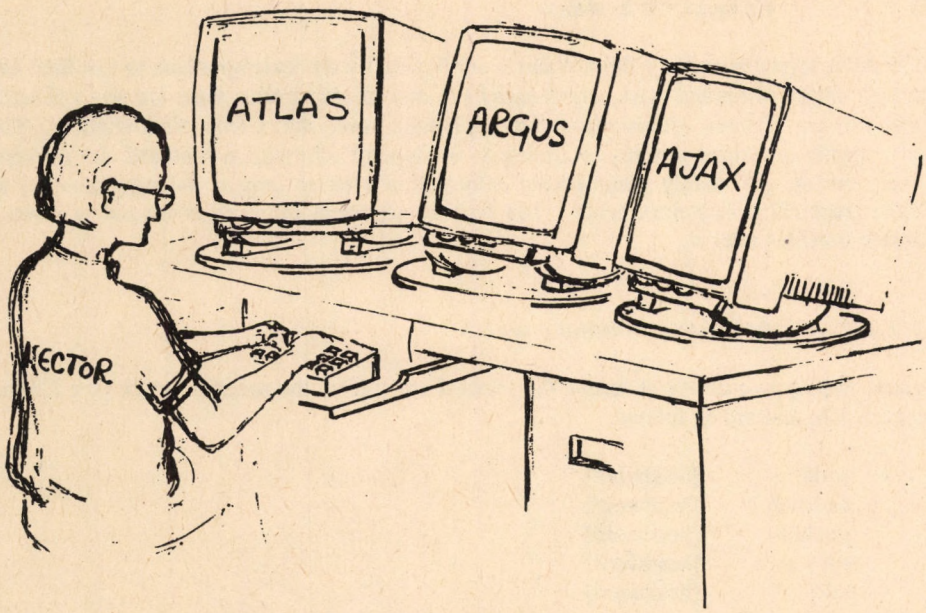


Figure 1: Atlas, Argus and Ajax monitors

This triple screen configuration allows the lexicographers to scan, sort and sense-tag the corpus data, and consult other reference sources, without losing track of the "shape" of the growing lexical entry. The cursor travels freely from one screen to another. Material may be cut from the corpus data in Argus and pasted in (as examples) into the appropriate field in Ajax.

3.1 THE ATLAS REFERENCE TOOLS

The left-hand monitor ("Atlas") is the reference screen. These are the tools which present the lexicographer with processed linguistic data, offering dictionary browsing facilities, statistics on wordform occurrences and collocational frequences, and a checklist of verb patterns. In addition, the lexicographers use the Atlas monitor for requesting information about texts in the corpus, or a progress report on the project, and for checking the consistency of their compiled entries; this is also the screen used for routine tasks such as document writing and handling electronic mail.

3.1.1 Commands in Atlas

The Atlas screen is entirely flexible and is configured by the lexicographers to suit their own tastes; unlike Argus and Ajax, which each have customized screens, there are no fixed access points to any of these commands, which are simply keyed into a Unix shell window. They fall roughly into three groups, according to the type of information accessed: (a) reference work entries; (b) mainly preprocessed information such as corpus statistics; and (c) the lexicographical policy documents. The type of argument for each command is given in square brackets after it.

3.1.1.1 Accessing reference works

These display in quasi-print format the selected entry from the specified work (see 2.3, and Figure 10), and are as follows:

cod8	[headword]
shorter	[headword]
pocket	[headword]
oxthes	[headword]
odq	[headword]

3.1.1.2 Mainly accessing preprocessed information

beth [verb-lemma]

This command produces information about the complementation patterns and transitivity alternations in which the lemma participates (see 2.4).

checkentry [headword]

This command checks the contents of three fields (semantic domain, register and style labels, and grammar) in a compiled entry. These may each legitimately contain only a specific set of character strings, and those which do not conform are easily identified. Later, other fields will be added to the list of those that fall within the scope of checkentry.

coll [wordform]

This command produces two sets of figures derived from statistical analysis of the corpus (see Church et al 1990a, 1990b and forthcoming): mutual information and t-score. These are obtained by analysing all the words that occur in a window of five words to the right of each target word, and (separately) in a window of five words to the left, and computing for both right and left environments those that co-occur with the keyword significantly more often than chance. The output of this command is shown on the right side of the screen in Figure 5.

Both MI and t-score draw the lexicographer's attention to words that tend to occur together, but they have slightly different emphases. The t-test takes account of the absolute frequencies of the each of the words in the corpus, so that the highest scoring items are not only strongly associated with the target word, they also tend to be very common words in their own right. MI, on the other hand, tends to pick out the more unusual items, which a lexicographer might have missed. The t-test often does better at drawing attention to the function words which co-occur with the target word; this can be particularly useful in studying syntactic points, for example verb complementation patterns. MI gives greater emphasis to content words, which is especially helpful as a tool for organizing semantic studies of the target words.

corpusdoc [text-title-abbreviation]

An abbreviation of the source document title appears alongside each corpus line. This command expands it to give fuller information from the corpus catalogue about the text it refers to (see 2.2).

incs [lemma]

This command opens a window into the citations file (see 2.1.2), known informally as the "incomings"; every occurrence of any form of the lemma in the citations database is offered to the lexicographer.

printentry [lemma]

This command outputs a printed version of the compiled entry in the format shown in Figure 35.

stats [lemma]

This command calls up a summary of the number of occurrences of the word in its various forms; as well as giving the frequencies of occurrence of each lexical form, stats shows how many occurrences of the word, and of each wordform, are tagged as a noun, verb, adjective, adverb etc. The output of this command is shown on the left side of the screen in Figure 5.

tally

This command computes the compilation to date and reports the situation, noting what entries have been compiled by which lexicographer, on what date; their length and complexity; the percentage of the wordlist compiled and of the corpus sense-tagged at the moment of enquiry, etc.

3.1.1.3 Accessing lexicographical policy documents

The policy documents are an on-line style guide, setting out for the team of lexicographers the exact way in which various types of linguistic data must be handled during the compilation. These commands open a text window on to the policy documents file at specific points; some examples are:

comps	(compounds)	
gram	(grammar)	
ids	(multiword items)	
punct	(punctuation)	
reg	(register)	etc. etc.

3.2 THE CORPUS SEARCHING AND TAGGING TOOLS ("ARGUS")

The centre monitor ("Argus") offers corpus data to the lexicographers, with a dialogue facility that allows them to specify conditions on the corpus searches.

These tools operate on raw linguistic data. At the heart of Argus is the 17.3 million-word corpus, tagged at present with text structuring features (e.g. those marking paragraphs, sentences and other text units, also typographical details, etc.) and grammatical features (wordclasses and some clause roles); the team is at present tagging the nouns in the corpus with fairly broad semantic features such as PERSON, PLANT, VEHICLE, GARMENT, FOOD etc. These are being drawn semi-automatically from an on-line thesaurus, but similar tags could of course be assigned from an on-line dictionary, using semantic taxonomies constructed from the parsed definitions, as described in Chodorow et al (1985), Byrd et al (1987) and Calzolari & Picchi (1988).

Argus uses these existing tags in the selection of concordance lines conforming to conditions set by the lexicographers, and displays these for sense-tagging; Argus also records the sense-tag assigned to each occurrence.

3.2.1 Commands in Argus

The command system in Argus and Ajax is very flexible. Some of the command buttons execute the command directly; others open up into a menu of further commands and options, possibly cascading to several levels; and others open a dialogue box into which the lexicographer must key some information.

The basic Argus layout is shown in Figure 2. On the screen there are two windows, upper ("Query") and lower ("Argus"), of equal size in this illustration, but the lexicographers can change this if they want to. Each has its own command line, which I shall describe in turn. Suspension points following some of the commands indicate the presence of a menu of options.

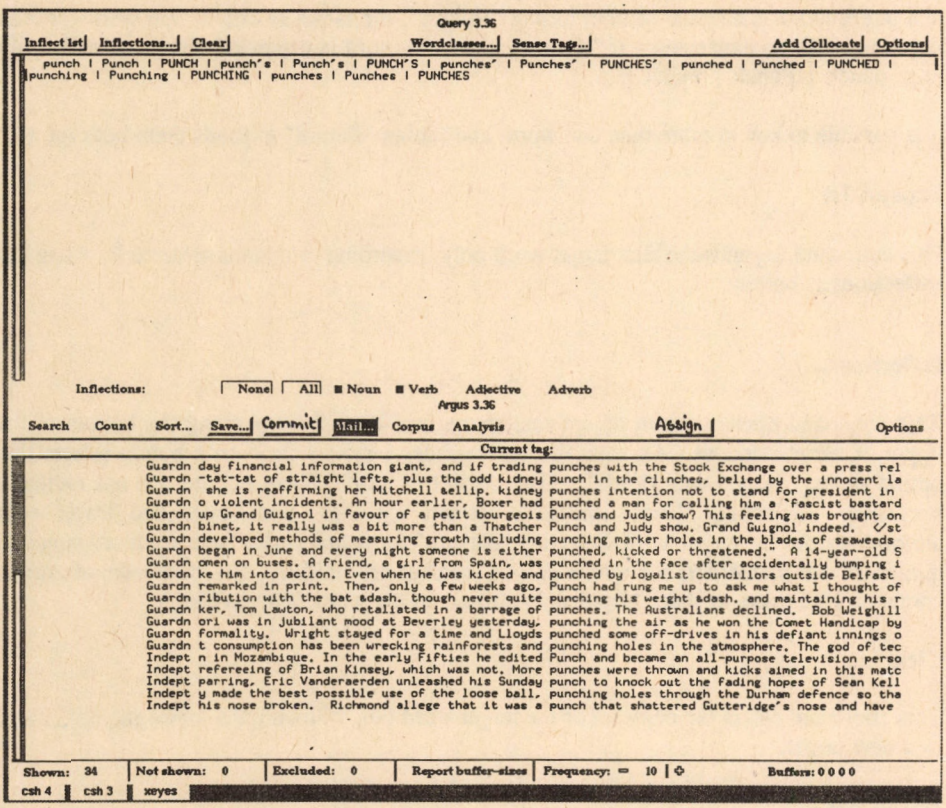


Figure 2: Argus screen

3.2.1.1 The "Query" window

The first three buttons in the command line (**Expand 1st, Inflections...**" and **Clear**, at the top of the screen in Figure 2, starting from the left) all operate on the same data: the word keyed by the lexicographer into the space below. This "target" word forms the focus of corpus searches.

The word may be "expanded" into all its possible forms, covering all the typographical alternatives for all the selected morphological inflections: in the screen shown, the noun and verb forms of *punch* have been expanded. The vertical bar separating the wordforms stands for "or".

If a target word is entered without hitting "Return", the target is exactly the word form as keyed in, with no expansion. If "Return" is hit, the word is expanded typographically only (e.g. punch | Punch | PUNCH).

It is possible to key in more than one word, and hitting "Return" expands them both (or all).

Expand 1st

This command expands the first target word only, according to options selected by using the **Inflections...** button.

Inflections...

This command opens a menu of options, shown in Figure 2 lying along the bottom of the upper "Query" window. As with all these menus, a mouse click on a button selects that option (here, "Noun" and "Verb"). The default is "All". These options are not mutually exclusive. Selecting "None" produces variant spellings, and upper and lower case alternatives, for the target word. Selecting "Adverb" for *punch* would have produced *punchly, punchlier* and *punchliest,* for which corpus matches are unlikely. (In the matter of corpus contents, I've come to share Napoleon's attitude to the word "impossible".)

Clear

This command clears the contents of the target word box, allowing the lexicographer to key in a new target.

Wordclass...

This command opens a dialogue box that allows the lexicographers to add further constraints to the search for target word occurrences, using the wordclass tags in the corpus. The default is "All". The options at present are "Noun", "Verb", "Adjective", "Adverb" and "Other". Soon, "Other" will be replaced by "Preposition", "Pronoun", "Article" etc. However, the primary constraint is the output of the expansion process: the search will be performed only for wordforms appearing in the target box.

Sense Tags...

This command constrains the search still further, by offering the lexicographer the chance to search only for those occurrences already tagged with specific sense-tags. It opens up into a list of the active sense-tags for the target word (see 3.4) and the lexicographer may choose them all, or some, or none. This is useful when you are looking for all the occurrences of one particular sense, or when you want to exclude, during the sense-tagging process, all the lines you have already sense-tagged. Here again, the default is "All".

Add Collocate

This command opens a new search condition area, allowing the lexicographer to key a target collocate word into the target box. This restricts the search to occurrences of the target word in the context of a specific collocate, or specific collocates.

All the options on the target word (*punch* in Figure 2) - **Expand 1st, Inflections..., Clear, Wordclasses...** and **Sense Tags...** - are repeated here for the collocate, and in addition the dialogue box Position allows us to specify either a position or a range of positions for the collocate location vis-à-vis the target word. In selecting these, the default is "-5,+5" (within a range of positions from five words to the left of the target word to five words to the right). The options here are fully flexible: "+1" would restrict the collocate search to the word occurring directly after the target word, "-20,+20" would greatly extend the search window, and so on.

The **Add Collocate** command functions, if required, on the wordclass tags alone; that is, it is possible to search for all occurrences of the target word with a preposition, or noun, or any other part of speech in a certain relative position. The words which match this condition are colour-highlighted in the resultant concordances, making it easy to see patterns, find phrasal verbs etc.

There is no limit to the number of collocate boxes the lexicographer may add. The relationship between the collocates is one of conjunction ("and"); that is, if I add the two collocates *rum* and *fruit,* each in its own collocate box, to the target word *punch,* Argus will find only concordance lines where both *rum* and *fruit* co-occur with *punch.* If I want to find all the lines where either *rum* or *fruit* co-occurs with *punch* I must enter both collocates into the same collocate box, with a vertical bar between them.

The collocate box can itself generate another collocate box (a collocate of the collocate). Positions in that box are relative to the parent collocate, not to the target word.

The facility to add collocates to both the target word and to collocates, together with the relationships of conjunction and disjunction that may be stipulated, make this search procedure extremely flexible (if rather complex and time-consuming).

Options

This command allows the lexicographer to reposition the Query window, and opens the following menu:

Move to Screen 0
Move to Screen 1
Move to Screen 2.

3.2.1.2 *The "Argus" window*

This window displays the result of a search and most of the commands relate to the concordance lines in the window. The command line with buttons **Search, Count** etc. lies at the top of the window (horizontally, centre screen, in Figure 2); the six boxes (**Shown, Not shown** etc.) at the foot of the window report on the search. I shall first describe the command buttons, starting from the left, and after them the search results boxes.

Search

This button starts a search, using input from the Query window. The resultant concordance lines start to scroll in the display window. The context of the target word is approximiately 50 characters to the right and 50 to the left; an abbreviation identifying the source text lies on the left of the line (in Figure 2, these indicate the British newspapers *The Guardian* and *The Independent*), while words matching the collocate specifications are colour-highlighted. The blank area to the left of the lines is where the lexicographers insert sense tags.

The lines appear in the order that the texts are accessed in the corpus. After a few weeks of experimentation, we decided that initially it would be most useful for the lexicographers to have the written texts sorted first on genre (with newspapers first, and fiction last), and within this primary sort, resorted on title, alphabetically; the transcribed spoken texts come last.

Count

This command simply counts the occurrences that match the search conditions, without displaying the concordances, and displays the total in the **Not Shown** box below the concordance window. It is much faster than **Search**.

Sort

This command takes the output of the **Search** command and sorts it according to several options. There are two sorts, a primary and a secondary (see Figure 7). Each offers the same options, viz:

Words At Right	=	alphabetically sorted on right context
Words At Left	=	alphabetically sorted on left context
Wordform	=	sorted on the wordform of the target word
Corpus	=	sorted in order of genre in the corpus
Sense	=	sorted on the sense tags that have been input

The default here is "Corpus" in both cases. Sorting on right and left context does not operate beyond the sentence boundary.

Save

This command puts the occurrences that result from a Search into a file according to further specification from the lexicographer. The options here are either to make a file of KWIC concordances as displayed on the screen, or to make a file of the full sentences in which these occurrences figure.

Commit

This command (often invisible in the screen dumps because of the colours on the monitor) saves the sense tags that the lexicographer has assigned to the corpus occurrences.

Mail

This command opens an instant hot line to the lexicographers' "minders", the Digital computer scientists who keep the software running and continue to add enrich it as the lexicography develops. **Mail** is a streamlined bug-reporting system.

Corpus

This command, called when the cursor is on a concordance line in the Argus display window, opens a window into the corpus. The options here are "Partial" (a window of 1,000 characters to the left and right of the target word) or "Entire" (the whole corpus).

Analysis

This command, called when the cursor is on a concordance line, opens a scrollable window where are displayed the wordclass tags and clause analysis data, including subject and predicate marking, that the sentence carries.

Assign

Here again, the label on the command button does not come out well in the screen dumps. This command assigns a designated sense tag (the one appearing at **Current tag** just below it) to a previously selected line or batch of lines in the display window,
thus allowing for batch tagging.

Options

This command opens up a menu of options. In addition to those for moving the Argus window to another screen, it includes several of direct interest in debugging operations, and the option to Quit, i.e. close down Argus in an orderly way.

Shown

is the first of the search results boxes lying along the base of the Argus window. In it is displayed the count of occurrences that match the search conditions, and for which KWIC concordances are being displayed.

Not shown

This command also gives the total of matches, this time for the output of the Count command, when the matches are displayed on the screen.

The other boxes are concerned with the running of the software, and normally used only by the computer scientists in the team.

3.3 THE DICTIONARY ENTRY EDITOR ("AJAX")

The right-hand monitor ("Ajax") houses the entry-building software; it is there that the lexicographers call up skeleton template entries and flesh them out into full entries.

These tools provide a structured environment in which the lexicographers compile lexical entries. This can be done from scratch, starting from a blank template, or a COD8 entry can be called up in Ajax for adaptation for the new dictionary-database in preparation.

The structure of the entries in Ajax is based on deconstructions (or parses) of the COD8 entries. Ten types of entry structure were identified, and the parses drawn up[5].

The types of entries for which parses were made are:

```
abbreviation_entry
affix_entry
biographical_entry
contraction_entry
crossref_entry hidden=FALSE
geographical_entry
given_name_entry
institution_entry
lexical_entry
nonassimilated_entry
```

5 This is the work of Rosamund Moon.

The least complicated of these (that for a foreign loan word) is given here, as an example:

```
nonassimilated_entry uid = ###.###.###.###
                     hidden = FALSE
                     ord = numlet(0)
                     sort = ()
free_text
pronunciation
variant_spelling*
decoration*
see_also_xr*
source
na_typology
wordclass_sequence+
derivative_sequence*
etymology
note*
```

where:

X+ = one or more X
X* = zero or more X

The parse (like all such) formalizes the structure of the entry-type it describes and defines. Here is one "nonassimilated" entry in print:

bonhomie /ˌbɒnɒˈmiː/ *n.* geniality; good-natured friendliness. [F f. *bonhomme* good fellow]

Figure 3: COD8 "non-assimilated" entry

The relationship between the lines of the parse and the elements of the entry is transparent enough.

This work informed the whole design of Ajax, and of course also allows Ajax to validate input data in certain fields, check that no essential part of an entry has been omitted, and order the entry components.

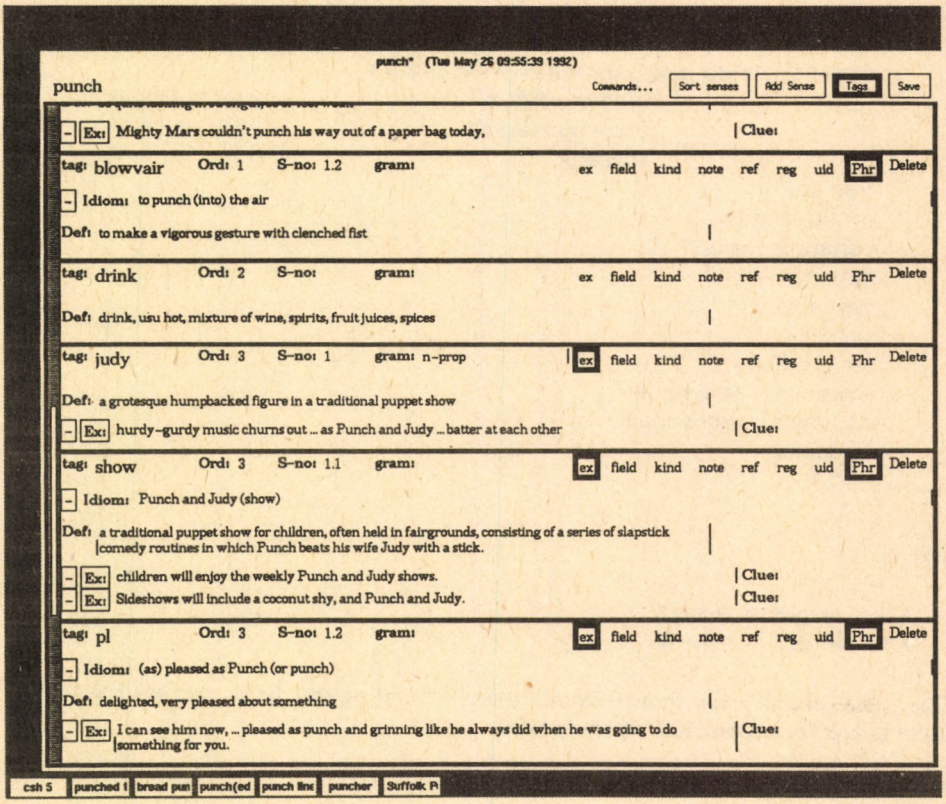


Figure 4: Ajax screen

3.3.1 *Commands in Ajax*

The Ajax screen shown in Figure 4 is known informally as "proto-Ajax". It is not a template for a full lexical entry, simply an outline format to be used for the basic structuring and compiling stages. We started off with a full dictionary entry template ("Ajax" proper), but this was so complex that, for all but the very short entries, no amount of manipulation would allow enough material on the screen at any given moment. We then reduced the number of fields and options, for the working draft template ("proto-Ajax"), and devised the "icon button" method of accessing other fields (see 3.3.1.2).

The design of "proto-Ajax", and indeed of full Ajax, is not yet complete. As the lexicography continues, and the computer scientists respond to requests from the lexicographers, the design is modified, and the operations get slicker and the screen more adapted to dictionary compiling. Proto-Ajax gives the lexicographers the ability to write a considerable amount of detail about each dictionary sense, and at the click of a mouse either

to increase the scope of the entry by adding another data field, or to close up the data fields and summarize on the screen a lot of senses at the same time, while even more of the entry is visible in quasi-dictionary format through the **Show** command (see 3.3.1.1). Throughout this paper, "Ajax" refers to the "proto-Ajax" that was used to construct the demonstration entry.

3.3.1.1 The Ajax command line

In Figure 4 the headword (*punch*) is entered in the extreme left corner of the command line at the top of the screen, where the headword normally appears in print dictionaries. Keying in a different word here automatically closes the current entry (with safeguards) and calls the entry for the new word, or if there is none, a blank template.

Most of the commands are accessed from a menu under the **Commands...**" box; the four most frequently used (**Sort senses, Add Sense, Tags** and **Save**) are buttons in their own right.

The command line contains the following options:

Commands...

This opens up a menu of commands:

Another Ajax
Odel Entry
Cod8 Entry
Full Ajax
Copy
Show
Alphabetically
Quit
Move to Screen 0
Move to Screen 1
Move to Screen 2

Another Ajax

This command brings up another blank template and holds both templates in the active mode, so that mnemonic tags from both entries are "official" at the same time in Argus (useful if you are writing entries for, say, *punch* and *punch up* at the same time, since Argus will not distinguish these occurrences in the corpus search).

Odel Entry

This command ("Odel" is the temporary name of the dictionary-database being compiled) calls up a compiled entry and displays it in Ajax format.

COD8 Entry

This command does the same for a COD8 entry (useful if it only needs tweaking to fit the new dictionary).

Copy

This command copies the current Ajax entry to a file.

Show

This command displays the current entry in something like print dictionary format. This is an excellent tool and one that is in constant use. It is very difficult to compile on-line without being able to take an overview of the entry very frequently. **Show** gives us the facility to do this.

Alphabetically

This command opens a dialogue box and invites the insertion of a keyword that will serve to alphabetize the entry within the headword list, if the form of the headword makes it impossible to do this automatically (e.g. to put "42nd Street" amongst the Fs).

Quit

This command exits Ajax, with appropriate precautions.

Move to Screen 0 / 1 / 2

The last three commands in the menu allow repositioning of the Ajax material (few lexicographers use this facility).

There are three other top-level commands in Ajax:

Sort Senses

This command (the next button in the Ajax command line, see Figure 4) sorts the senses according to the numbers entered at **Ord** (= ordinal number) and **S-no:** (= sense number) in each sense frame, and displays the newly sorted version on the screen. It is not always necessary to be able to see the very latest ordering displayed in the Ajax screen, and the lexicographers, unwilling to add even a few seconds to the time spent waiting, rejected automatic re-sorting of the entry each time a sense number was changed.

Add Sense

This command adds another sense frame at the bottom of the Ajax template. Each frame holds the contents of a dictionary sense. It is possible to add an empty frame at a particular point within the entry by positioning the cursor and striking an initialized function key.

Tags

This command transmits the mnemonic tags to Argus (the corpus search tool), at which point the tags become official, each with a unique identity in the database. Argus responds by expanding the **Sense Tags** button in the top line of the Query window (this may be seen in Figure 12) and listing the officially recognized mnemonics that will be accepted as sense-tags against corpus lines. Any mnemonic tag in Ajax can of course be eliminated, or changed. This is done by making the changes in Ajax, and toggling **Tags** off, and on again; Argus responds by listing the new set of tags. Block de-assignment and reassignment of tags to corpus lines is also possible.

Save

The **Save** button saves all the Ajax data inserted since the last save.

3.3.1.2 Buttons in the Ajax sense frames

I shall explain here the options represented by these buttons; their use will become apparent as their role is described in sections 4.2 and 4.3.

tag

Beside this label is a text field into which the lexicographer keys the mnemonic chosen for that particular sense (in Figure 4, for instance, 'blowvair', 'drink', 'judy' etc.).

Ord and **S-no**

These also label text fields, which hold numbers assigned to homograph headwords (**Ord**) and sense divisions within the entry (**S-no**).

gram

This text field receives the abbreviations for parts of speech and other grammatical information to appear in the dictionary.

The other (icon) buttons all operate in the same way: one mouse click repositions the button against the left margin of the frame at the correct point in the entry vis-à-vis the other data

in the entry (for instance, the various **ex** and **Idiom** buttons in Figure 4), and opens a text field for lexicographers to key data into, together with a "minus" button which will kill the field, if they change their mind later. If a second text field of the same type (for instance, **ex** or **field** or **note** or **ref**) is needed within one sense frame, this can be called up by clicking on the first. Another mouse click on the **ex** or **field** (etc.) icon at the top of the frame closes any active fields of that type; if there are more than one, it iconifies them together in one button.

The data types, which correspond to items in traditional dictionary entries, or to information to be held in a database although perhaps not to appear in a print dictionary, are as follows:

ex.

dictionary example

field

semantic field or domain marker

kind

a shorthand name for a field that holds listings of compounds of which this headword is an element (e.g. in the Ajax entry for *punch*, in the frame for the "blow" sense, the **kind** field might hold *kidney punch* and *rabbit punch)*; the contents of this field will not necessarily appear in the dictionary

note

from lexicographer to lexicographer, not for publication

ref

cross-reference to another entry

reg

register, style etc. marker

uid

displays the unique identity number of that sense in the database

Phr

allows the insertion of a multiword lexical item of which the headword is one element (idioms and phrasal verbs are identified separately within this area)

Delete

deletes the whole sense frame instantly

3.4 AJAX - ARGUS INTERCOMMUNICATION: SENSE TAGGING

The link between Ajax and Argus is the system of sense-tagging. During the compiling process, the lexicographer assigns to each occurrence of the headword in the corpus a tag relating it to one of the senses of the lexical entry. The words for which entries are being compiled during this phase of the project have on average between 500 and 1,000 occurrences in the corpus. A method had to be found of tagging each occurrence on its first (and possibly only) appearance on the Argus screen - there was no question of sense-tagging the corpus after the entry was fully compiled, as that would have doubled the lexicographical work. At the same time, the lexicographers had to be free to change their minds as often as they needed to, at any point in the compiling, over the number and order of the dictionary senses of the target headword. And they had to be able to do that without needing to re-tag the corpus occurrences that had already been sense-tagged.

This is the system that was devised: when lexicographers begin to draft an entry on the Ajax screen, they give each sense a mnemonic tag; for instance, one sense of *punch* might be tagged 'drink', one 'judy' and one 'show', as in Figure 4. To each tag Ajax assigns a unique identity number (incorporating the headword in it), and will therefore not accept two identical mnemonics within the same headword entry. The compiler hits a key to transmit these tags to Argus. Beside each KWIC concordance line on the Argus screen is a "sense-tag box" (in Figure 2, this lies in the empty column to the left of the KWIC concordance lines); Argus will accept in that box only "official" tags, i.e. mnemonics which have been duly transmitted from Ajax (such tags are visible in Figure 23). When two senses are merged in the lexical entry (in Ajax), the concordance lines may be retagged in a block in Argus. When senses are moved around in the Ajax lexical entry, this has no effect on the corpus sense-tags in Argus, as the unique identity numbers remain the same. When a tag in the entry is renamed with a different mnemonic, this is transmitted to Argus, and the corpus mnemonics are duly changed.

I shall now describe how, with these tools at my disposal, I drafted an entry for the word *punch*, tagging its senses into the corpus, and I shall try to show how the various Hector tools function at each point in the compilation.

4. ENTRY COMPILING AND SENSE TAGGING

The work of a lexicographer is essentially one of analysis (discovering, understanding, structuring and recording the facts about the word to be described) and synthesis (from this ordered set of facts, selecting those appropriate to the dictionary-database being compiled, and setting them out in the way most helpful to the typical user of the eventual dictionary). The lexicographical work goes into a database recording the grammar, meaning, and use of words in current English, which will in due course constitute the core of a major new Oxford dictionary.

There is no canonical method of compiling dictionary entries from corpus data. All of us do it differently. But although techniques vary, the essential operations are the same for everyone. This diversity has proved an enriching and challenging factor in the Hector project, since each lexicographer has had something new to contribute to the demands made on the software. Perhaps the greatest challenge of course is the requirement that every operation should be carried out with maximum speed.

4.1. GETTING A FIX ON THE WORD

The first step is to get acquainted with the word: to see how it behaves, to begin to distinguish large chunks of meaning in its semantics, to see what wordclasses is belongs to, to start to notice what are its commonest forms, to learn its collocation patterns. In the Hector project, getting a fix on the word before starting to compile an entry involves studying the corpus statistics (on the Atlas screen), scanning the corpus (in Argus), and consulting other dictionaries (in Atlas).

4.1.1 Corpus statistics

The two commands available are described in 3.1.1.2. The command "stats punch" calls up a summary of the occurrences of *punch* in its various lexical and morphological forms (on the left half of the screen in Figure 5). It could be useful to note that the word occurs so often with a capital P (no doubt as the name of the British humorous magazine, now alas defunct, and in the expression "Punch and Judy"), because when it comes to searching for blocks of same-sense citations in the concordances, this form should prove easy to find, and pull out a rich haul of occurrences to be sense-tagged in a block.

The command "coll punch" (on the right half of the screen) produces an extract from the collocational frequencies file. This command operates on wordforms only (hence the disparity in the frequency of occurrence between the output of **coll** and that of **stats**). **Coll** looks at collocational frequencies of the wordform 'punch' not the lemma *punch*.

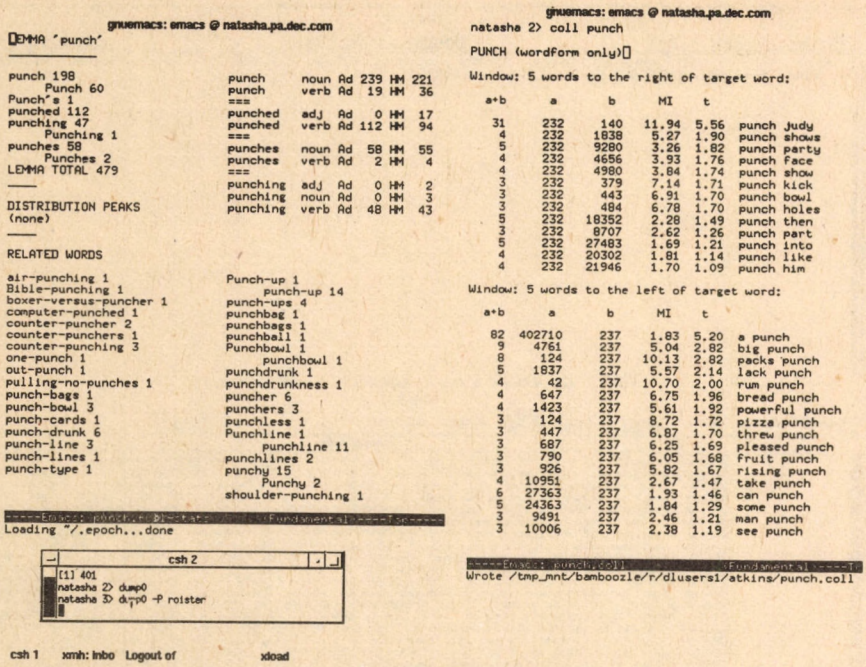


Figure 5: Response to "stats punch" (left) and "coll punch" (right)

In Figure 5, the collocates of 'punch' are listed in t-score order. In this case the t-test picks out as most significant among the left collocates the cooccurrence of *a* and *punch* (a point with some syntactic significance). By contrast, the highest scoring items by the MI test are *rum* with *punch* and *packs* with *punch*, both drawing the lexicographer's attention to important multiword items: the compound "rum punch" and the idiom "to pack a (powerful etc.) punch". Given the difference between the two tests, it is all the more remarkable that both of them agree that 'judy' is far and away the most significant right collocate of the wordform 'punch'. Clearly, the Punch and Judy show continues to play a major role in English culture.

4.1.2 *Scanning the corpus*

The first step is to tell Argus the target word - in this case, *punch*. This word is keyed in to the target box in the Query screen (see Figure 6), and the **Inflections** key is hit to bring up the **Inflections** options (central on the screen). Since I want to scan through the range of corpus lines, I hit the "Noun" and "Verb" buttons, identifying these wordclasses as active for

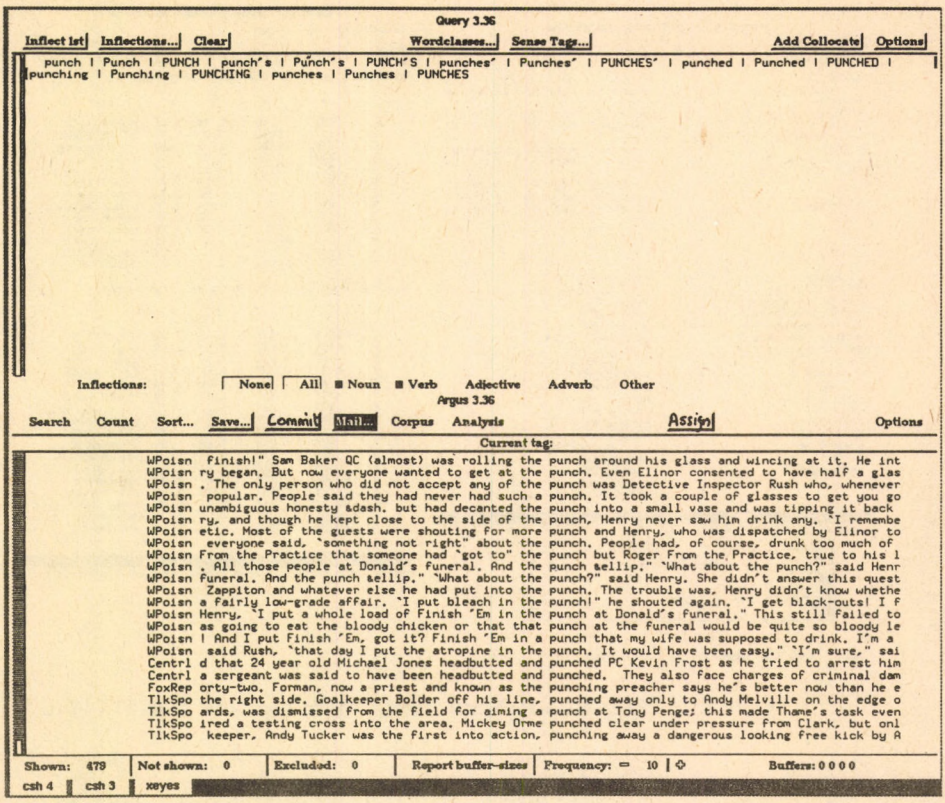


Figure 6: (Argus) *punch* expanded, with KWIC concordances, unsorted

the expansion process. When I hit the **Inflect 1st** button, Argus expands the target word as a noun and as a verb, and the various lexical forms appear (as in Figure 6), separated by "or" bars. These forms are the target for the Argus searches, and will remain so until I change them.

There is now the option of simply counting the occurrences that match the Search condition, by hitting the **Count** button in the command line of the lower Argus screen; the result of the count is given in the **Not Shown** box, but no matches are displayed. This is much quicker than pulling up concordance lines from the corpus, but in this instance I know what the count would be (the **stats** command has already reported 479 occurrences of the lemma), so I hit the **Search** button and the KWIC concordances for *punch* start to scroll.

At this point, unconstrained by any Search conditions, the occurrences appear in the order that the texts are accessed in the corpus. As the corpus lines scroll past, I start to notice

points about the context of *punch* in its various uses. I can stop the Search scrolling when I am ready to move to the next operation, or I can leave it until all instances of *punch* are collected. In either case, I can scroll up and down through the concordance lines that the Search has produced.

Normally, at this point I wish to start structuring the data a little. I hit the **Sort** key on the Argus command line, and the options are displayed.

I choose to have the lines sorted first on the basis of right context ("Words At Right"), and within that primary sort on wordform (see Figure 7). Experience suggests that for a word like *punch*, this is the most useful first sort.

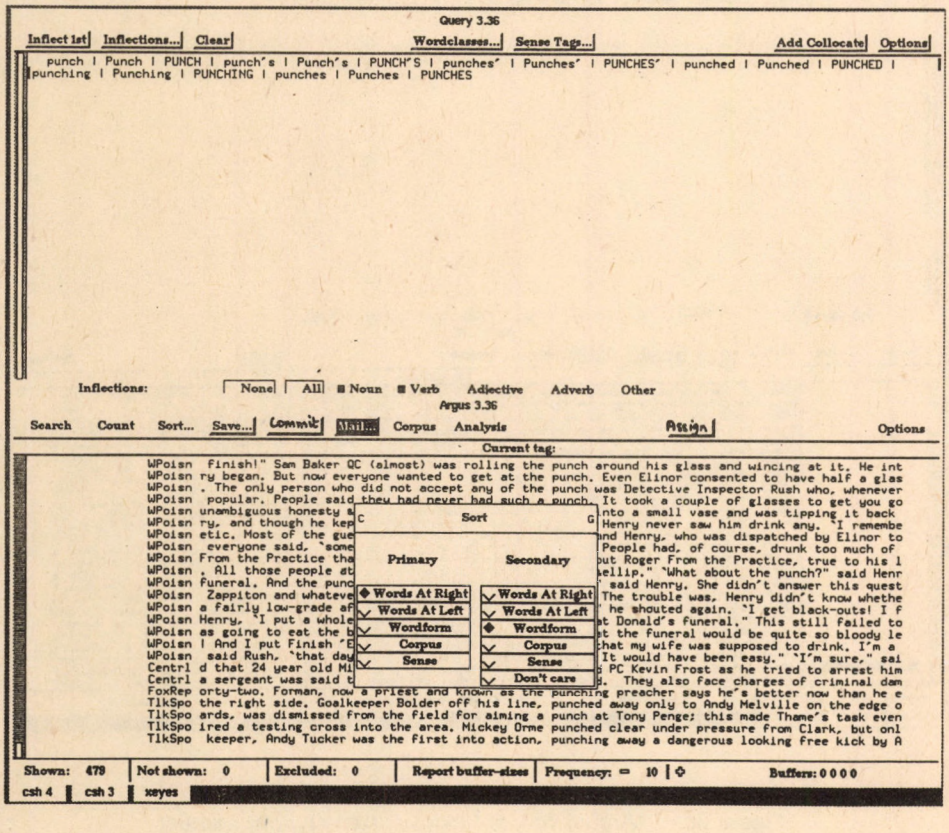


Figure 7: (Argus) showing Sort options

The sorted lines appear almost instantaneously (see Figure 8) , and - still trying to get the "feel" of the word - I scroll through them. The right context sort brings out very clearly some of the patterning highlighted by the statistics output from the coll command (see 4.1.1). "Punch and Judy" is clearly a common collocation.

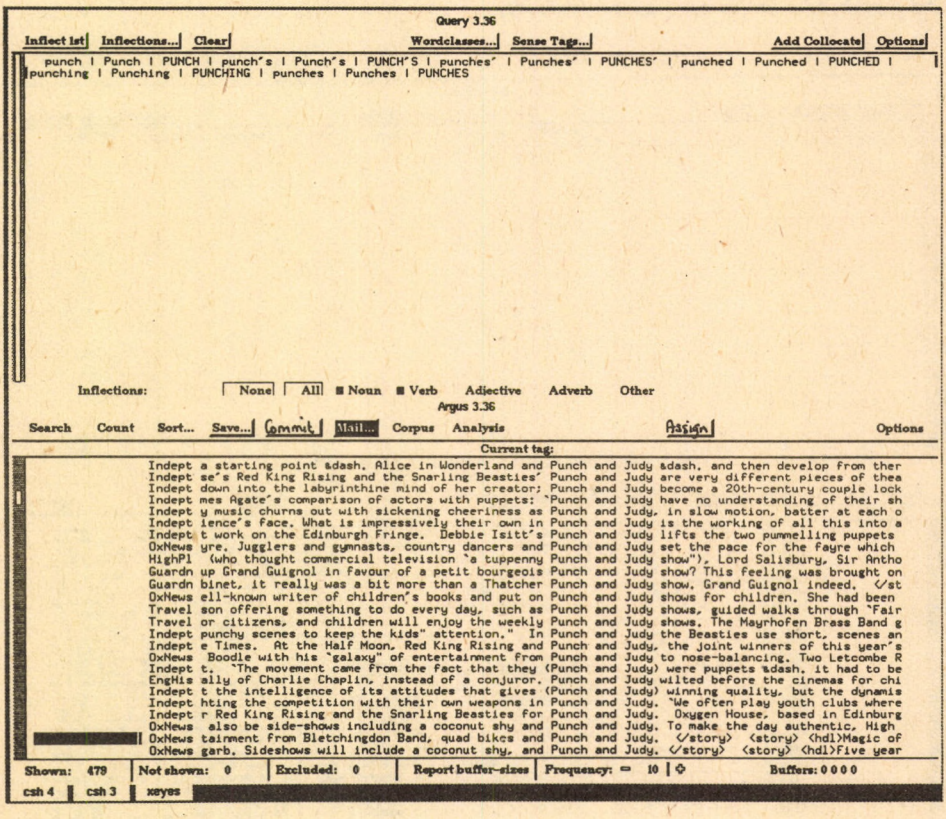


Figure 8: (Argus) concordances sorted on right context

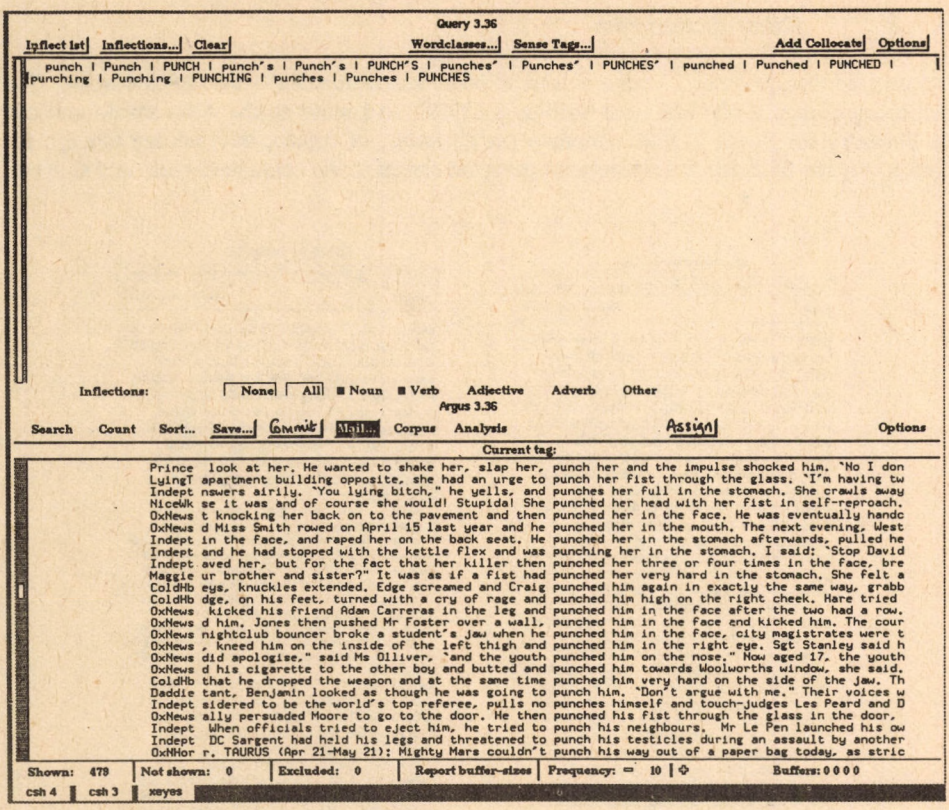


Figure 9: (Argus) typical complements of verb *punch*

Further down the lines alphabetically sorted on right context we come to the pronoun objects (*him* and *her*) of the verb *punch* (see Figure 9). People are punched on the nose, in the face, in the stomach, in the mouth - indeed, in most body parts. This common patterning must show up in the finished entry.

Other patterns are noted, as the scrolling continues, and gradually the word starts to become familiar, its profile begins to emerge from the mists of the language, distinctive features are discerned, and I am nearly ready to make a start on the first draft of the entry.

4.1.3 Other dictionaries

Before I do that, however, I want to look at other lexicographers' views of this word. The dictionary closest to the one I am writing is COD8, so I move to the Atlas screen and call up the entry for *punch* in that dictionary (much faster, of course, than leafing through the book). Figure 10 shows the response in the Atlas screen to the commands **cod8** and **shorter**.

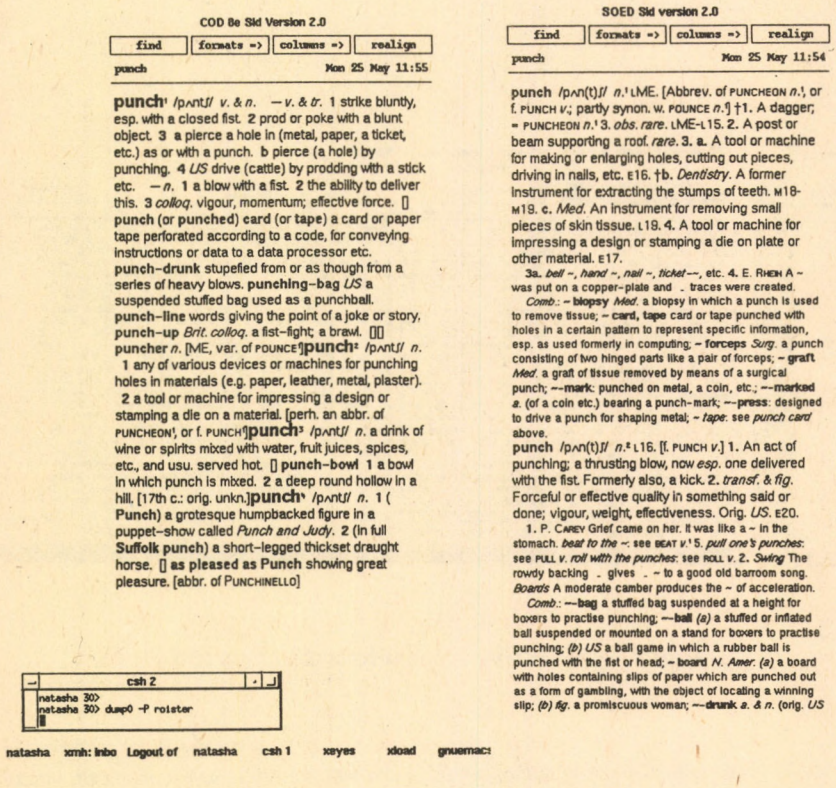


Figure 10: Response to "cod8 punch" (left) and "shorter punch" (right)

4.1.4 Verb patterns

Lastly, to remind myself of the potential of the verb, I key the command "beth punch", in the Atlas monitor. I am offered excerpts from sections 8.2 and 9.1 of the verb listings, showing the various patternings that apply to verbs of similar semantic content to *punch*. In the extract below, the verbs *cut*, *swat* and *punch* itself are used to exemplify these patterns:

```
***  (+ hole)
     to    VERB      a hole in <B>
     to    cut       a hole in sth

***  (+ resultative)
     to    VERB      <B>       resultative
     to    cut       sth       open

***  (+ bodypart)
     to    VERB      <C>'s     bodypart
     to    punch           sb's        nose

***  (bodypart possessor ascension)
     to    VERB      <C>       on  bodypart
     to    punch           sb  on  the nose

***  (conative alternation)
     to    VERB      at   <B>
     to    swat      at   st

***  (zero nominal)
     to    give  <B> a    VERB-NOUN
     to    give  sb  a    punch

     to    get   a   VERB-NOUN      on  bodypart
     to    get   a   punch          on  the nose
```

4.2 *THE FIRST OUTLINE ENTRY AND SENSE-TAGGING THE CORPUS*

A first sketch is made in Ajax, where a few broad senses are identified and the mnemonics assigned. It is clear, both from the first scan of the corpus, and from the entries in other dictionaries, that there are certain distinct senses to be identified for the noun and the verb *punch.*

I start by setting up mnemonic tags for these senses, both as noun and as verb, in the Ajax template on the right-hand screen (see Figure 11 where the mnemonics appear in large bold type against **tag** on the left of the screen). Among these initial senses are: striking a blow with clenched fist (verb and noun, with mnemonics 'blowv' and 'blown' respectively); making a hole in something ('pierce'); the tool that does this ('tool'); vigour and effectiveness ('vigr'); the fruit drink ('drink'); and the puppet ('judy').

The design of this database calls for a separation of noun and verb senses; wordclass tags in the corpus allow a search for noun occurrences and verb occurrences separately. It is therefore practical to group the verb senses together and the nouns together from the start of the lexicography. I try to do this when establishing my first sense mnemonics.

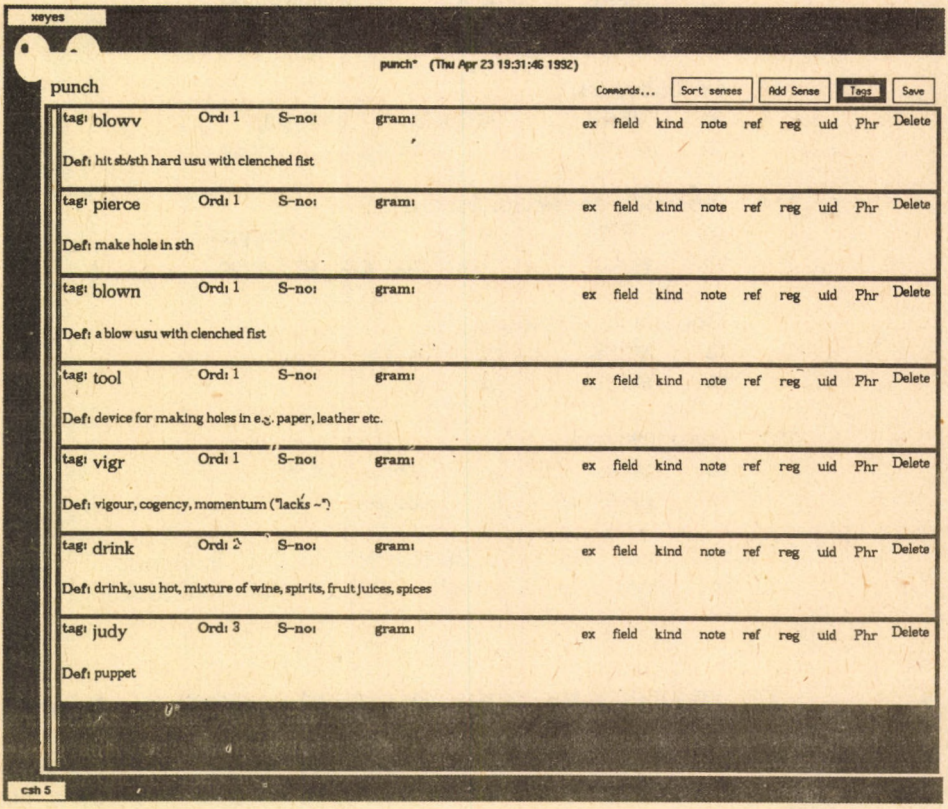


Figure 11: (Ajax) start-up template with the first few *punch* mnemonics

Figure 11 shows the first few senses sketched out. It is not necessary to enter Ajax tags in any particular order, since it is always possible to insert a new sense frame at any point in the outline entry, or to kill one by hitting the **Delete** key within the sense frame. Once the sense numbers (S-no: in Figure 11) are entered, Ajax knows the ordering of the draft entry, and will reorder the sense frames on the screen when the **Sort senses** command button is hit.

When **Tags** is hit in Ajax, the mnemonics become official in Argus. On the centre screen (see Figure 12), Argus displays the sense tags available for use, and I start working through the occurrences of *punch* in the corpus, gradually building the draft dictionary entry, and sense-tagging the 479 corpus lines. The objective, of course, is to tag them as far as possible in batches, and the tools are designed to facilitate this.

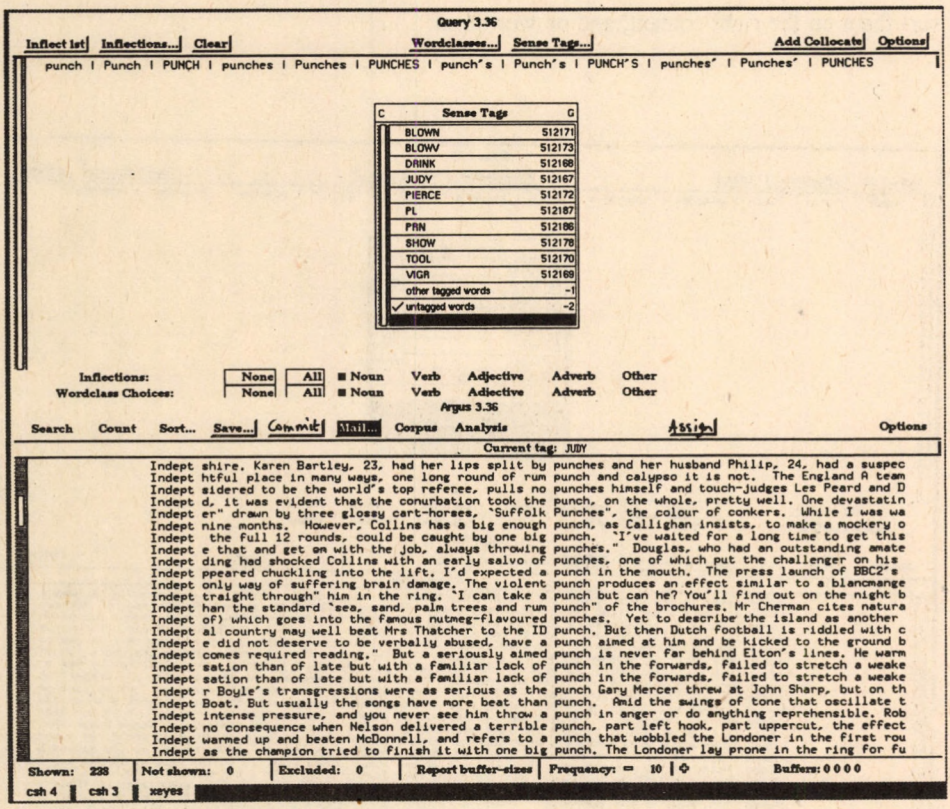


Figure 12: (Argus) showing "official" sense tags received from Ajax

First, I decide to look only at noun occurrences. This involves resetting the target of Argus's search procedure, by hitting the **Clear** button in the top command line of the Query window (see Figure 9), which empties the text area. After keying in 'punch' again, and altering the Inflections options to read "Noun" only, a mouse click on **Inflect 1st** expands the wordform 'punch' as a noun (see Figure 13). I also put a condition on the search by hitting the

Wordclasses button, which opens a further band of options **Wordclasses** at the foot of the Query window, and opting for "Noun". This tells Argus to read the wordclass tags in the corpus and select only occurrences of *punch* tagged as a noun (two taggers have pre-tagged the corpus, neither wholly reliably, and at this point Argus offers any word that either of the taggers has identified as a noun; some verbs creep in under the net). I hit **Search** in the Argus window and the corpus lines start to scroll (see Figure 13).

I sort them on the right context, and on wordform.

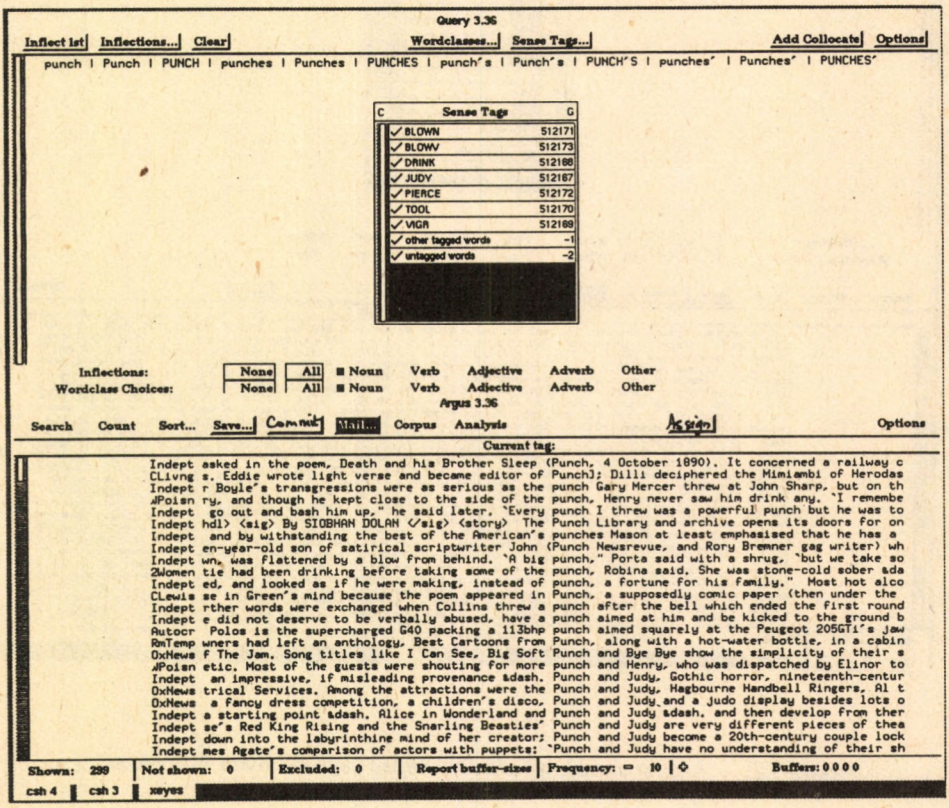


Figure 13: (Argus) noun *punch* lines sorted on right context and wordform

It immediately becomes apparent - from phrases such as "editor of Punch" - that I need a mnemonic for *Punch* as the name of the magazine, and I add that ('mag') to the Ajax template. When I toggle the Ajax **Tags** button, Argus accepts the mnemonic and adds it to the displayed list.

On scrolling through the nouns, I realize that a search for "Punch and Judy" will bring up a batch to be sense-tagged at a stroke. This is done by adding a collocate to the search constraints.

A mouse click on the **Add Collocate** button in the top command line of the Argus window opens a new search condition area; this may be seen below in Figure 14. I key in 'Judy' (which needs no expanding) and change the **Position** default setting to +2. Argus should display all concordances where the word 'Judy' appears as the second word after the target word. A mouse click on **Search** produces 28 lines that match the search conditions (*Punch* tagged as a noun, with *Judy in* position +2). At this point, the lines being correctly sense-tagged, I want to save these tags into the corpus text. To do this, I hit the **Commit** button in the command line of the lower half of the Argus screen. Tags which have been committed can be changed later, if need be.

I note the phrase "Punch and Judy show", which occurs six times, and want to hold it as a phrase within the draft entry. I adapt the Ajax draft entry accordingly, by inserting at the appropriate point of the entry another sense frame to hold it; by giving it the mnemonic "show" and clicking on **Tags** to make that official; and by clicking on a **Phr** button and keying in "Punch and Judy show" in the text field that appears. I also add a working definition, and an ex (example) field. To complete this tiny part of the entry, I return to the corpus lines in Argus to find a good example. Reading these more carefully, I realize that "Punch and Judy show" is often abbreviated to "Punch and Judy" (e.g. "Sideshows will include a coconut shy, and Punch and Judy"), and I adapt the entry accordingly. These sections are partly visible in Ajax format in Figure 4, and a later version of them appears as 3:1.1 and 3:1.2 in Figure 17.

I then return to the Argus window, to sense-tag the "Punch and Judy" lines. Those in which 'Punch' and 'Judy' refer to individual characters or puppets will receive the 'judy' tag, while those referring to the puppet show will get tagged as 'show'. To do this as economically as possible, I make 'show' the current tag, by keying it into the **Current tag**: field above the corpus lines. After that, the tag may be assigned to designated lines by simply blocking them out and hitting the **Assign** button. In this way, all the "Punch and Judy show" lines are tagged with the mnemonic 'show'. When I am sure that the tagging is correct, I hit the **Commit** button in the Argus command line and these sense tags are written into the corpus.

Now it is time to adapt the dictionary draft in Ajax by bracketing the word "show", since it is an optional element in the phrase, and I decide to hold an example of each form for the time being. To gather corpus examples, I scroll down through the Argus lines until I find

a likely-looking citation - "children will enjoy the weekly Punch and Judy shows"; I block this out in the Argus screen, move the cursor to the correct spot in Ajax, and paste it in to the dictionary entry as an example. I do the same again with an example of "Punch and Judy" in the sense of "Punch and Judy show", noting for future reference the curious syntax of the line selected: "Sideshows will include a coconut shy, and Punch and Judy".

Back in Argus, I block out all the corpus lines visible on the screen, change the **Current tag** to 'judy' and assign the tag with one mouse click to all the remaining "Punch and Judy" lines (those already tagged with 'show' are not changed). I "commit" these tags to the corpus, and look for an example of *Punch* used to designate the puppet, or the character. The line "as Punch and Judy, in slow motion, batter at each ..." catches my eye, but is incomplete. In order to find a coherent citation I need to go back to the corpus for a fuller context. This is quickly done,by hitting the **Corpus** button in the Argus command line, and the corpus context is displayed (see Figure 14). From the displayed corpus text, I block out the example I want,

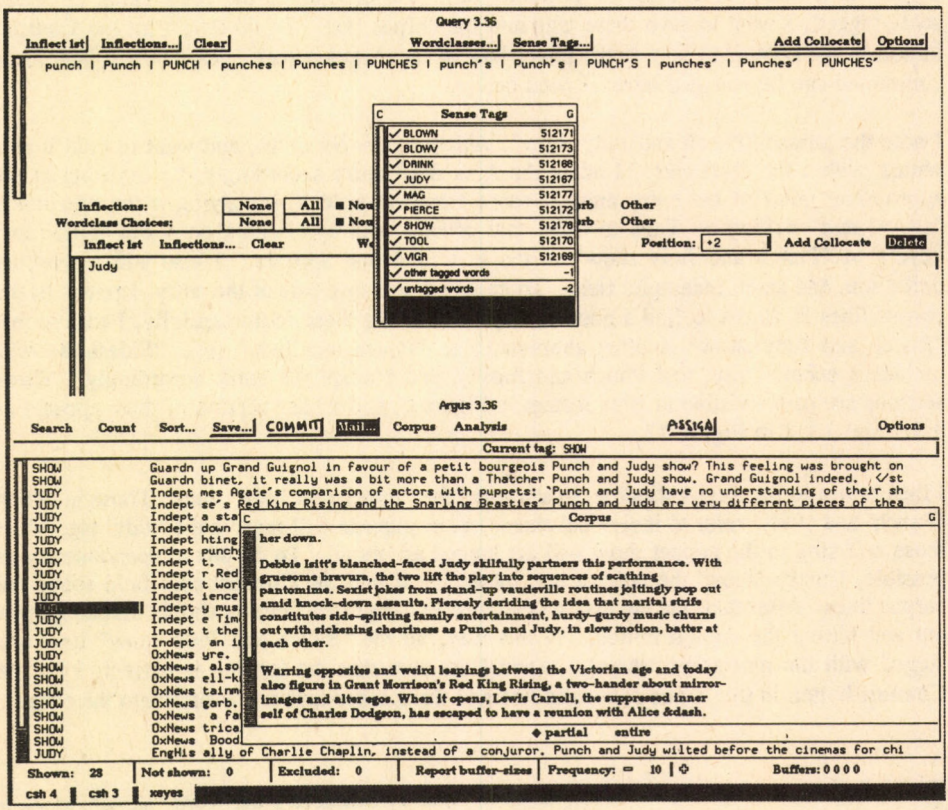


Figure 14: (Argus) collocate *Judy*, and extended corpus context shown

transfer the cursor to the Ajax screen and paste it into the ex field in the correct section. At this point, I number the senses in that section, which will be the third homograph headword, as its "Ord 3" numbering shows.

There will still be some corpus occurrences of 'Punch' which refer to the puppet, and others referring to the magazine. I decide to clear up that section of my draft dictionary entry, and return to the Argus screen. I reset the search conditions, this time looking for capitalized 'Punch' as a noun, with no specified context. (I kill the **Add Collocate** box containing 'judy', shown in Figure 14 by hitting the **Delete** button.) I also change the "on/off" toggles on the Argus Sense Tags list (Figure 15) so that Argus will display only occurrences that have not yet been sense-tagged. This facility to select according to the committed sense tags, as well as the other tagged features, greatly accelerates the lexicography. The output of this search is shown in Figure 15.

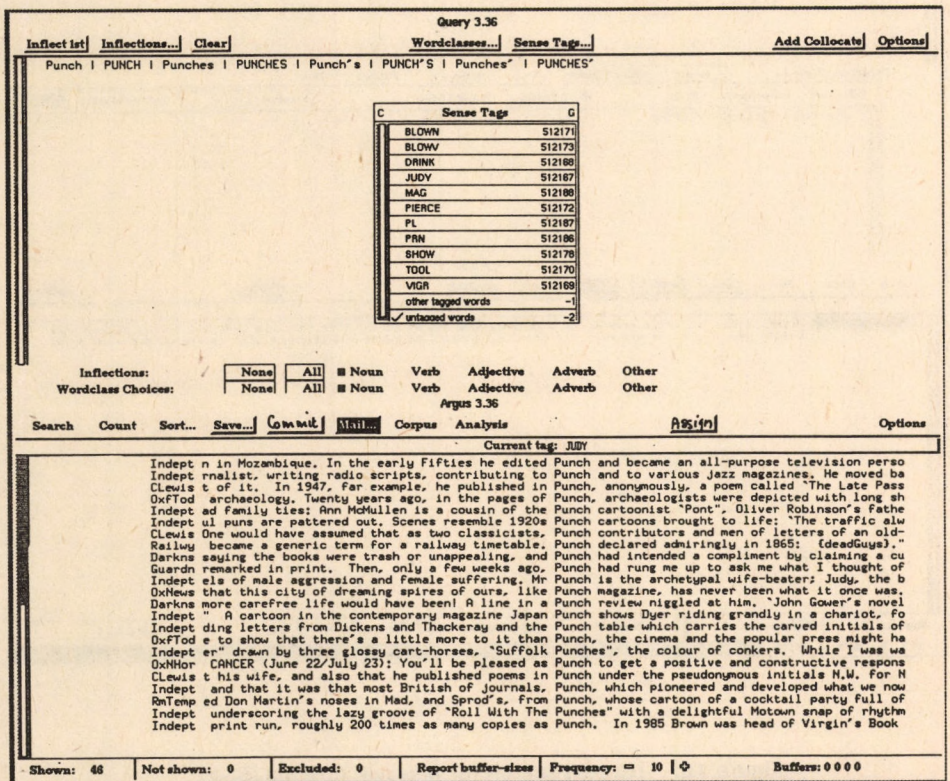


Figure 15: (Argus) capitalized *Punch*, noun, non-sense-tagged lines only

The corpus reminds me of the existence of the phrase "as pleased as Punch". I add a sense frame to the Ajax entry, initialize the mnemonic 'pl', and tag the single relevant corpus line on display. However I suspect that in this phrase, 'Punch' may appear without its capital letter. I therefore set up an Argus search for the word 'pleased' in the left context of 'punch' (see Figure 16, where the monitor colours identifying the "active" corpus line unfortunately show black in the screen dump) once again asking for untagged lines only, and finding only two, which I tag. I expand one to the corpus context and cut and paste an example for the draft entry.

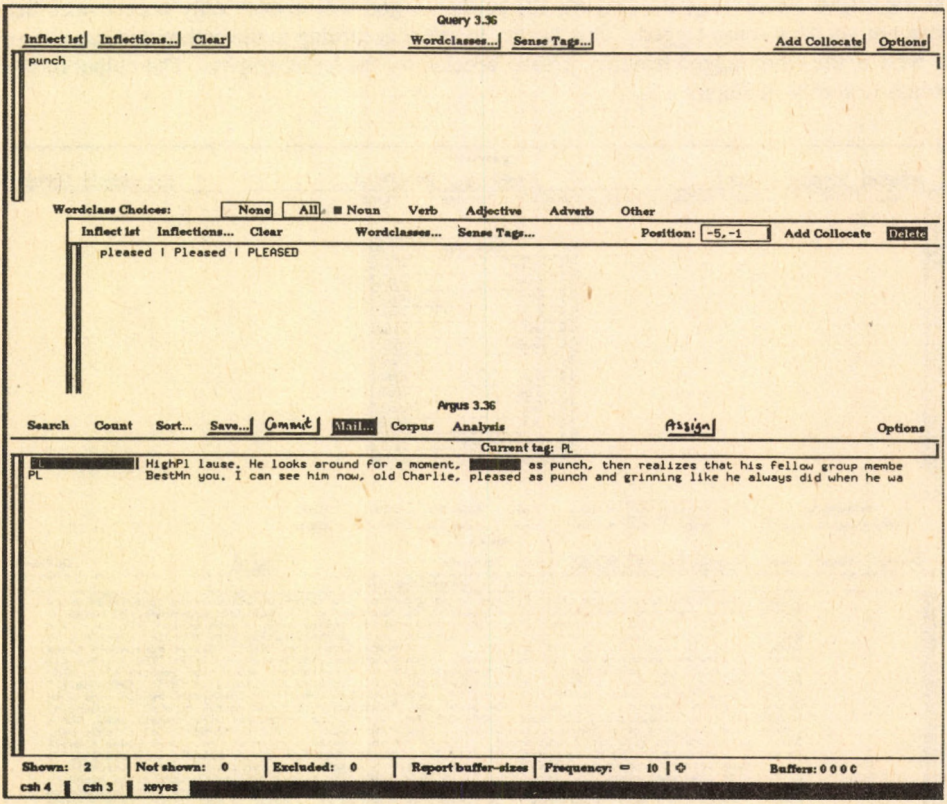


Figure 16: (Argus) search for *pleased* in range -5,-1

With this little section of the entry nearing completion, I hit **Show** in the Ajax **Commands** menu, and bring it up in pseudo-dictionary format, moving it to the Atlas screen (see Figure 17).

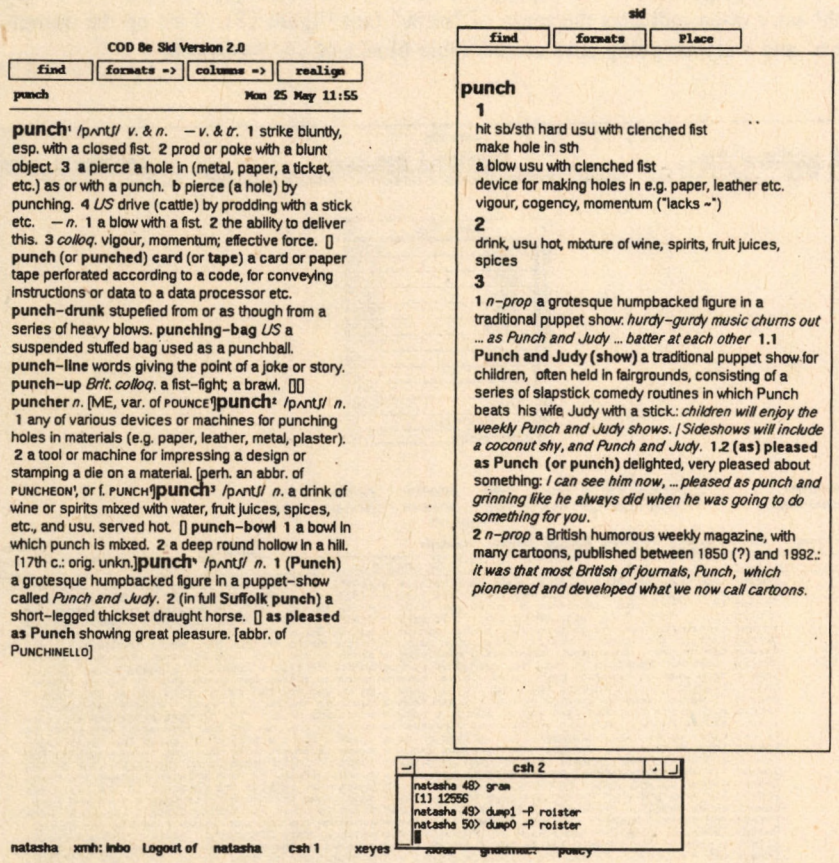


Figure 17: (Atlas) COD8 entry (left) and embryonic new entry (right)

The bulk of the entry for *punch* (on the right of the screen) is as yet unformed, but a small part of it is gradually emerging. I leave the COD8 entry there, as the comparison is always useful.

Before moving on to compiling another part of the entry, I ask Argus to count the number of untagged noun-tagged occurrences remaining: 238. I sort these on right context and scroll through them, looking for inspiration for a new search, to no immediate avail; I re-sort them first on wordform and second on right context, and notice that there seem to be a lot of adjectives modifying *punch*. I sort again, this time on left context, and realize that the phrase "a punch" very often indicates the sense of "blow" (see Figure 18). I set up the current tag as 'blown' and start assigning it to considerable blocks of concordances.

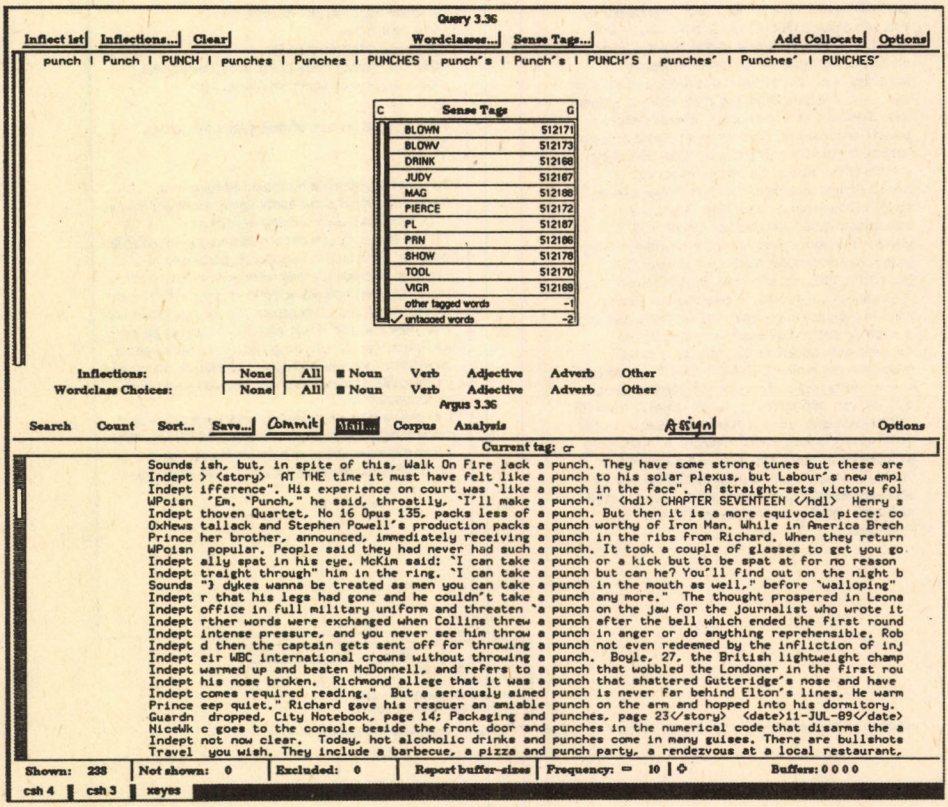


Figure 18: (Argus) the pattern "a punch"; "Add Collocate" box is closed

While doing this, I notice the phrases "to pack a punch" and "to throw a punch", and set these up as idioms in Ajax with the mnemonics 'pack' and 'throw'; the idiom "not to pull any (or one's) punches" is also recorded in Ajax with a mnemonic of 'pull'. Since 238 lines are too many to sense-tag individually, or even in blocks, I decide to set up more specific searches. In this way, I pull up 10 lines which I can block-assign as 'pull'. A similar technique accounts for 14 instances of "throwing punches" (all manifestly from commentaries on Boxing Matches, and I note 'Boxing' in the domain field of that section of the draft entry in Ajax). A third pass through the untagged corpus lines looking for wordforms of the verb *pack* in the immediate environment of *punch* produces 11 lines to tag with the 'pack' mnemonic (see Figure 19, where the monitor's colour-highlighting of the specified collocate is blacking out the occurrences of *pack*).

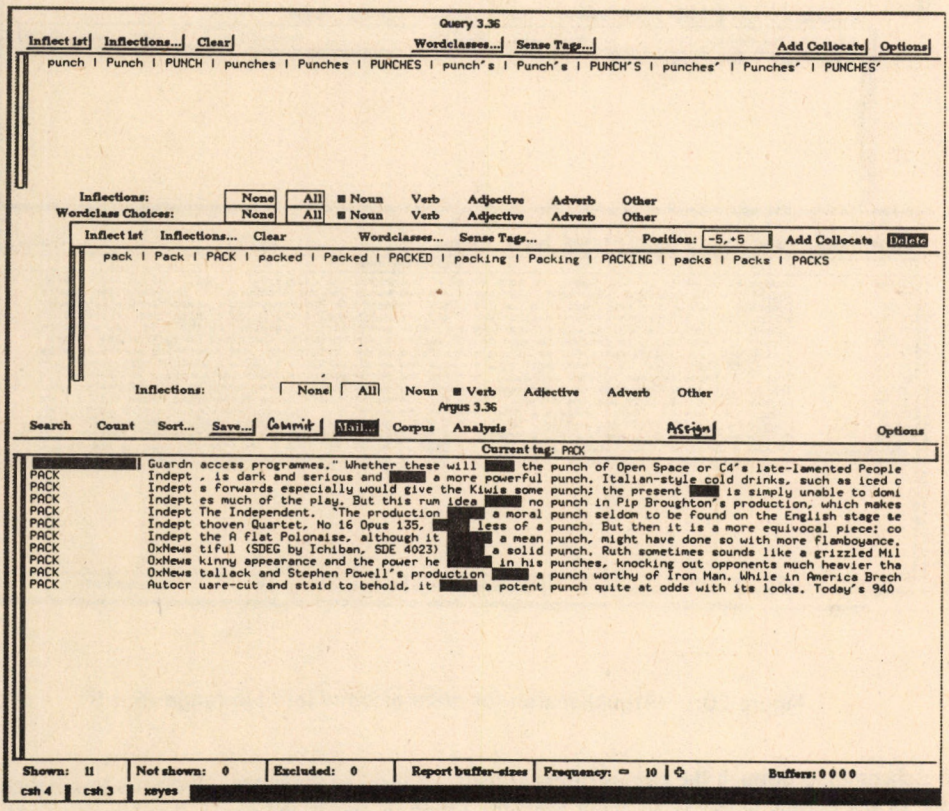


Figure 19: (Argus) search for *pack* in range -5, +5

Remembering a number of instances of *hot* collocating with the 'drink' sense of *punch*, I call them up, and tag six lines together. Checking back to the output of the coll command (see Figure 5), I set up a search for collocates *pizza, bowl, party, rum* and *fruit*, and glean another score or so of tagged lines (see Figure 20).

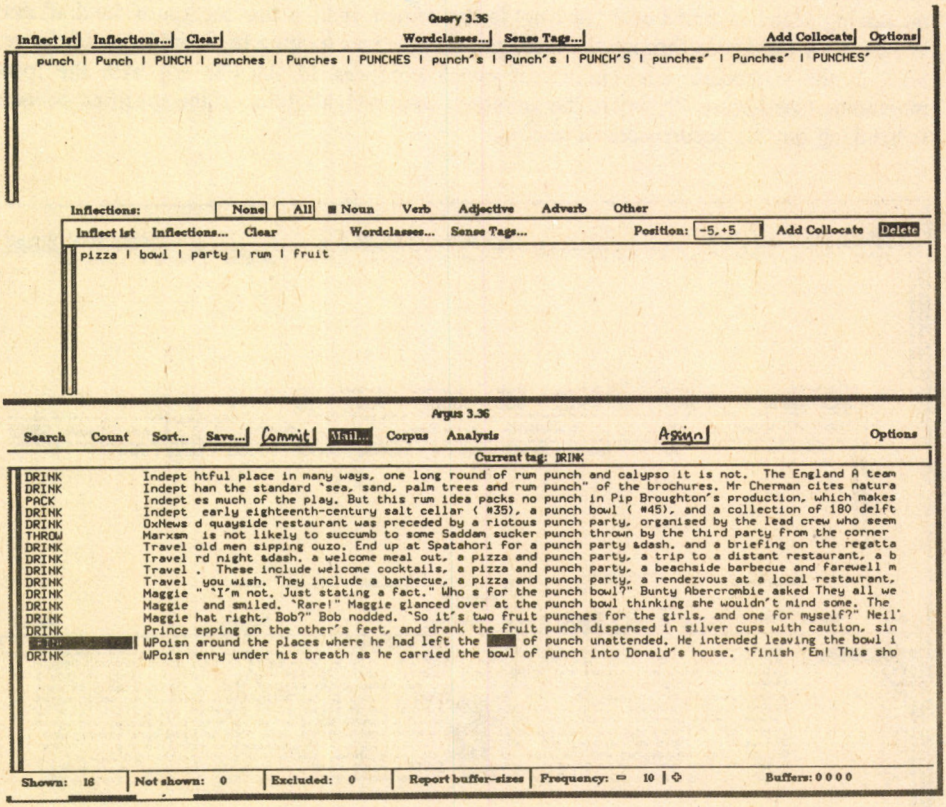


Figure 20: (Argus) search for *pizza* or *bowl* (etc.) in range -5, +5

Beachcombing through the corpus offers at least the thrill of the chase. It occurs to me that one of the corpus works is all about someone trying to poison the punch that his wife is to drink; I resort the lines in corpus text order and manage to tag 24 occurrences in one block that way. Another of the texts in the corpus contains a discussion about the drink *punch* - another block to tag together.

As a bonus, sorting the corpus according to source text points up the fact that the four citations from *The Angler* magazine show the compound *bread punch*. I raise a new Ajax template by hitting **Another Ajax** in the **Commands** menu (see 3.3.1.1), and keying a new (compound) headword 'bread punch' as a response to the "Which word?" query. A blank template appears. I insert a tag there (see Figure 21), together with a brief definition and a note about the single source text where this usage appears, then hit the **Tags** toggle, transmitting it to Argus. Argus now has as live and valid sense tags all the *punch* mnemonics and the new *bread punch* mnemonic. I tag the four concordance lines in Argus with 'brp'.

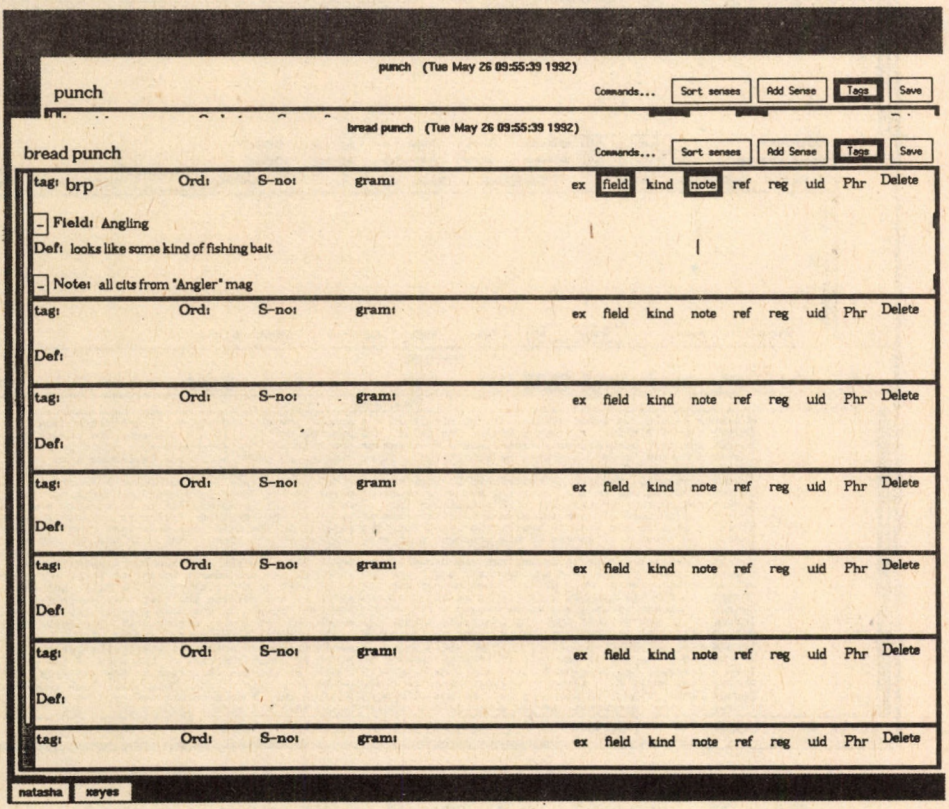


Figure 21: (Ajax) new entry for compound headword *bread punch*

I decide to look for occurrences of *punch* followed immediately by a preposition (in a hunt for "punch on the nose / jaw" etc.). The search conditions in Argus allow for search on wordclass tag alone, without any lexical item specified. This proves to be more fruitful, and 107 concordance lines (see Figure 22) are found. Not all are correctly wordclass-tagged, but most are - the **Other** selection in the **Wordclass Choices** box pulls out everything that is not tagged Noun, Verb, Adjective or Adverb. Later Argus will be able to search on the tags for other individual wordclasses such as preposition, pronoun, etc.

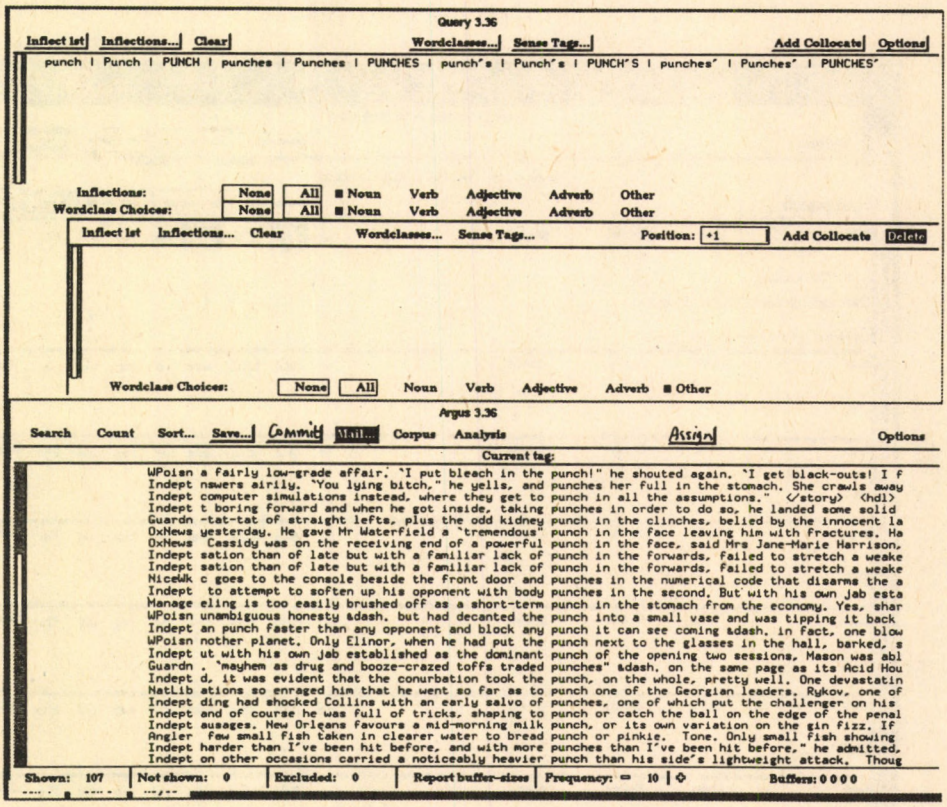


Figure 22: (Argus) search for "Other" wordclasses at position +1

I realize that there are still many untagged noun lines like "a punch in the face" or "a punch on the jaw". I therefore set up a search condition in Argus to pull out untagged concordances where *in* or *on* occurs within four words after *punch*, and Argus finds 30 matching lines, most of which I can sense-tag 'blown' (= "blow: noun") quickly, in blocks.

Wishing to tag the corresponding verb lines, I set up a search condition in Argus to pull out untagged concordances where *in* or *on* occurs within four words after *punch* verb occurrences, and Argus finds 55 matching lines (see Figure 23). It is clear that some semantic feature tagging on corpus nouns (e.g. BODYPART) would allow me to pull out large numbers of "assault" and "boxing" occurrences of *punch*, both noun and verb.

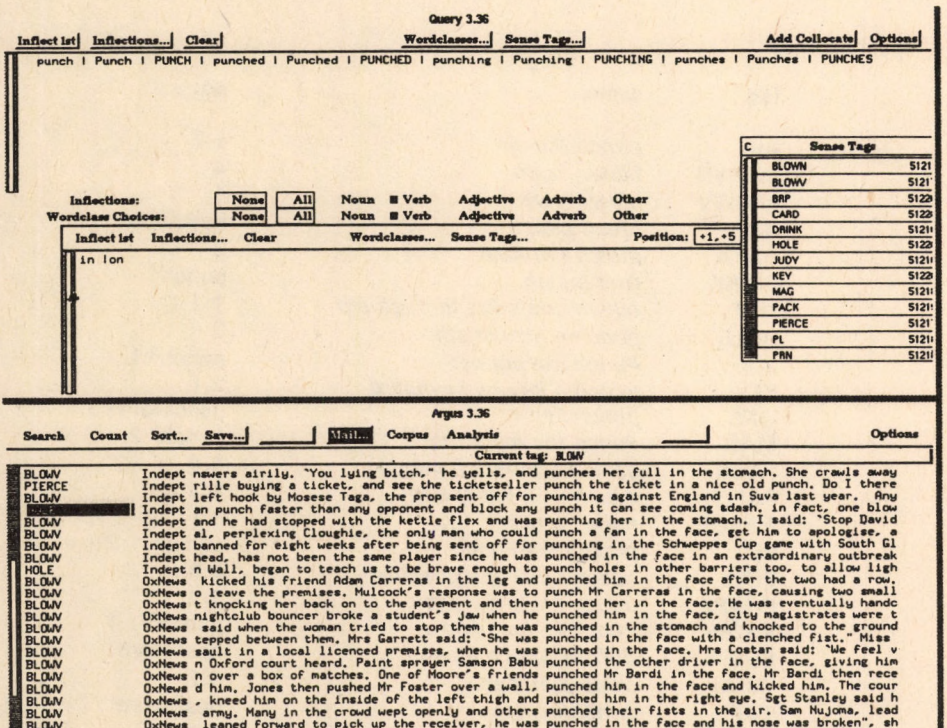


Figure 23: (Argus) *punch* verb with *in* or *on* in Range +1, +5 showing assigned sense tag "blowv"

Meanwhile, I notice other senses of *punch* for which I have not yet got tags established, e.g. "can be punched into the numeric keypad", for which I set up a new mnemonic. The compound *punch(ed) card* requires a separate headword entry in the database; this is made, and this compound is given a mnemonic.

At this point Argus is working on 26 active sense-tags, belonging to three headwords all concurrently active in Ajax (*punch, punch(ed) card* and *bread punch*). The tags are as follows:

MNEMONIC SENSE TAGS
for senses in entry shown in Fig. 35

tag	sense	no.
AIR	*punch the air*	1.2
BLOWN	*blow - noun*	4
BLOWV	*blow - verb*	1
BRP	*bread punch*	headword
DATA	*punch (in) data*	2
DRINK	*fruit punch*	punch[2]
FIST	*punch one's fist through sth*	1.1
HOLE	*punch a hole in sth*	3
JUDY	*Punch the puppet*	punch[3] 1
KEY	*punch a key on keyboard*	2.1
LINE	*punch line*	headword
MAG	*Punch the magazine*	punch[3] 2
PACK	*pack a (powerful) punch*	4.4
PAPER	*punch ... a paper bag*	1.3
PIERCE	*punch a ticket*	3.1
PL	*pleased as Punch*	punch[3] 1.2
PULL	*doesn't pull any punches*	4.5
SHAPE	*punch out a hexagon*	3.2
SHOW	*Punch & Judy show*	punch[3] 1.1
SUFF	*Suffolk punch (horse)*	headword
SWAP	*swap punches*	4.2
TAPE	*punch(ed) tape*	headword
THROW	*throw a punch*	4.1
TOOL	*tool for making holes in sth*	6
TRADE	*trade punches*	4.3
VIGR	*lacks punch*	5

Figure 24: List of "active" mnemonic sense tags for *punch, punch(ed) card* and *bread punch*

I set up the condition "Untagged lines" only, and hit **Count** in Argus, to learn that 223 lines remain to be tagged (see Figure 25).

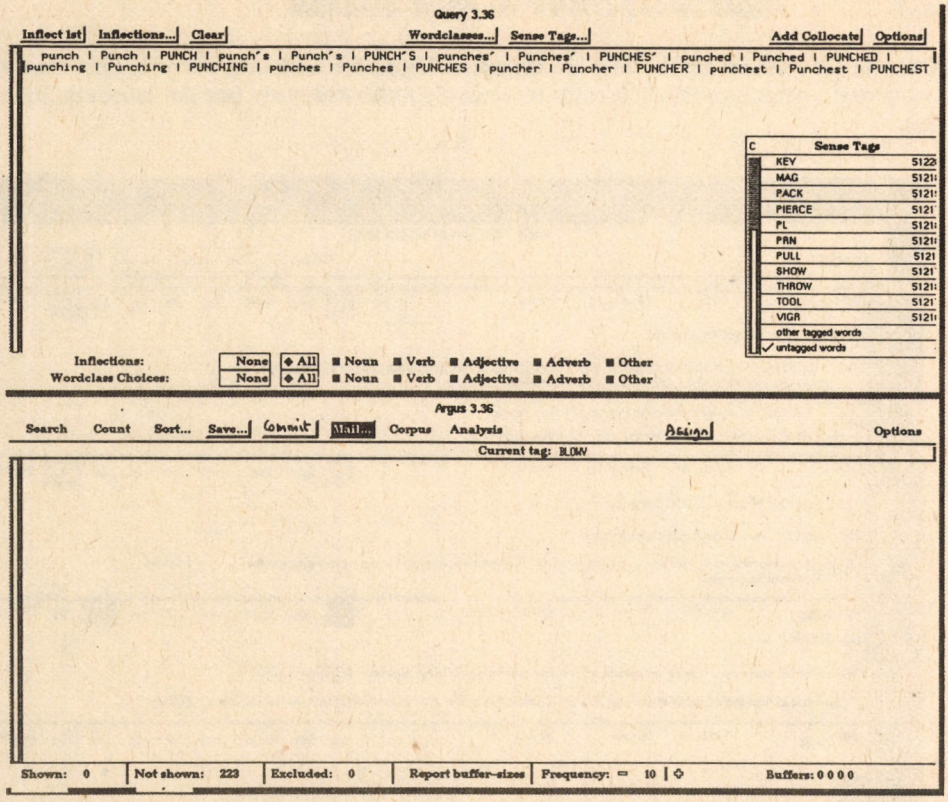


Figure 25: (Argus) response to search for all lines still without sense tags

Halfway through the tagging process the lexicographer's job satisfaction is at its nadir. (And *punch* is not a highly frequent word.) I scroll through these lines, looking for inspiration. How can I pull up large blocks of the same sense? I wish Argus could already find possessives and pronouns, since this would bring up all the lines like "punched her on the nose" and "punched him". I note with vague interest how often someone's name occurs as the object of the "assault" sense (mainly from newspaper texts, of course), and wish Argus could search on proper nouns, which, although tagged as a class, are at this moment still subsumed under **Other** wordclasses. I think that if we already had the semantic features tagged on nouns I could pull up all the +HUMAN nouns after *punch* and collect a whole senseful of citations that way.

I call up all the lines with *hole* as a collocate, and tag these, and similarly all the *punch(ed) card* concordances. Moving back and forth between Argus and Ajax, I finally complete the tagging procedure, adding en route headword entries for *punched tape, punch-up* and *punch(-)drunk*, bringing the sense-tag count up to 29.

4.3 FROM DRAFT ENTRY TO FINAL VERSION

Once all the relevant corpus lines are tagged in Argus, the dictionary entry in Ajax must be completed. At present the draft entry is extremely rough and ready (see the extract in Figure 26).

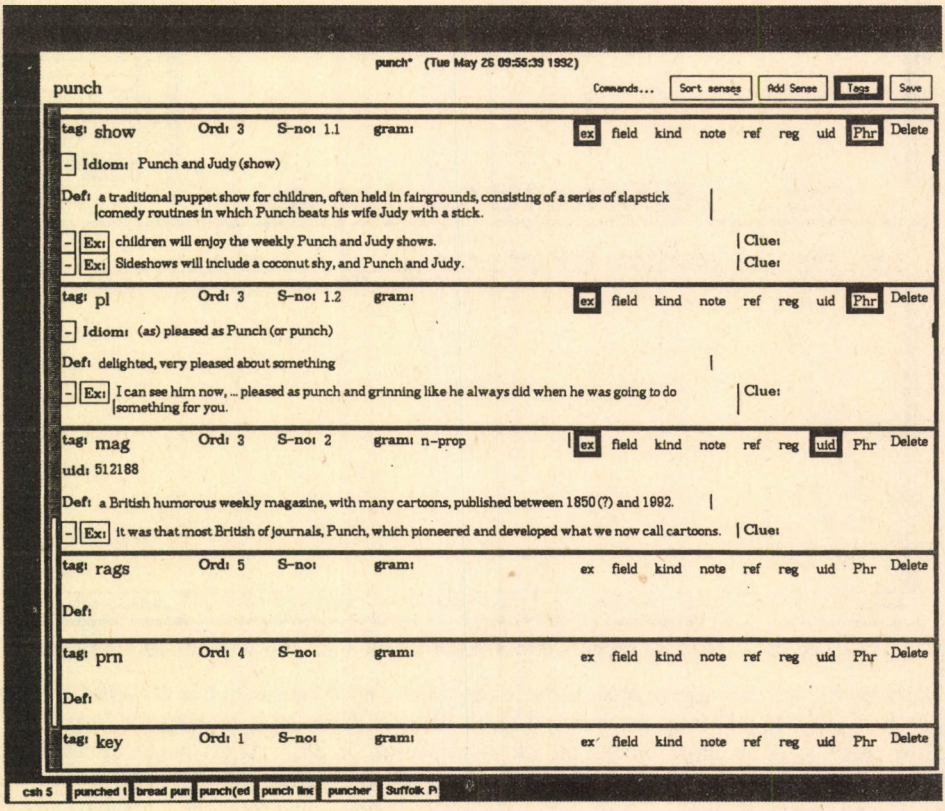


Figure 26: (Ajax) extract from *punch* early draft entry

All the senses are there, with their mnemonic tags, but there is no coherent ordering, and few if any completed definitions. Almost all the dictionary senses require exemplification, and most of the other information (register and domain labels, cross-references etc.) is missing. Entries for the currently active related compounds with headword status have been iconified and lie along the base of the screen.

The first task is to order the senses in Ajax, to get them as near as possible to what will be the final dictionary-database order. This is where the **Show** function in the **Commands** menu

comes into its own: an entry like *punch*, which (with related compound headword entries) has 29 active tags, will overflow the Ajax screen many times over. It is vital to be able to see the whole entry during the next part of the process. This is achieved by means of the **Show** command.

First, however, I make an initial attempt at numbering the senses and ordering them within the entry - or, rather, entries, for the "Ord" numbers in Ajax are used to distinguish homograph headwords. The provisional policy for this database is to treat as separate headwords homographs which are cognitively discrete, those for which an act of scholarhip is needed to link them etymologically or in any other way. Thus, the *punch* group will have three homographs (the main noun/verb, the drink and the Punch and Judy character).

Before I start on systematic sense numbering, I have to have some idea of what the contents of the entry are. I hit **Show** in the **Commands** menu of Ajax, and a pseudo-dictionary entry appears (see Figure 27).

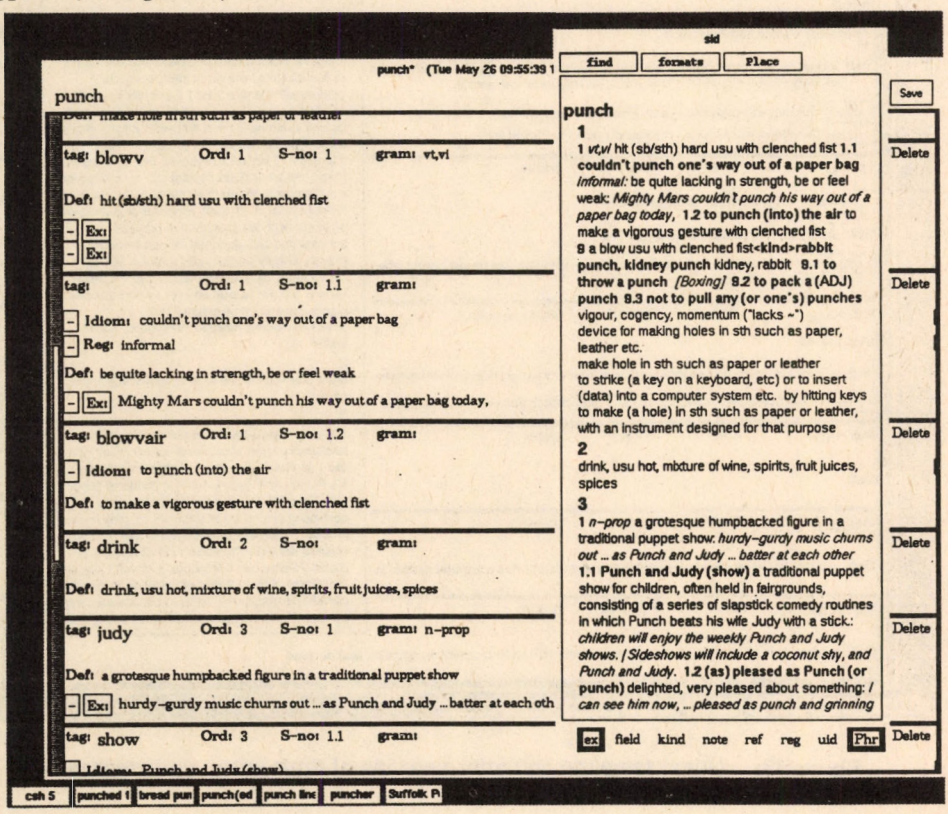


Figure 27: (Ajax) early draft entry in Ajax and print format

The three main homographs stand out clearly: the third is beginning to look reasonably solid, but the second lacks everything but a definition, and the first is a jumble of odd facts. Since the sense-numbering in Ajax is not complete (as is apparent in Figure 27), there are many sense numbers missing from the first major section (number 1) in the pseudo-print format; however, the senses themselves are there, as may be gleaned from a careful study of the content of the draft entry in Figure 27.

I use this overview to help me order the senses, and I also key in "cod8 punch" to bring the COD8 entry up in the Atlas screen, for reference. (I note in passing that COD8 marks *punch-up* as British English, and I add that to my draft entry for this word.)

I number the verb senses and call up another **Show** version (see Figure 28).

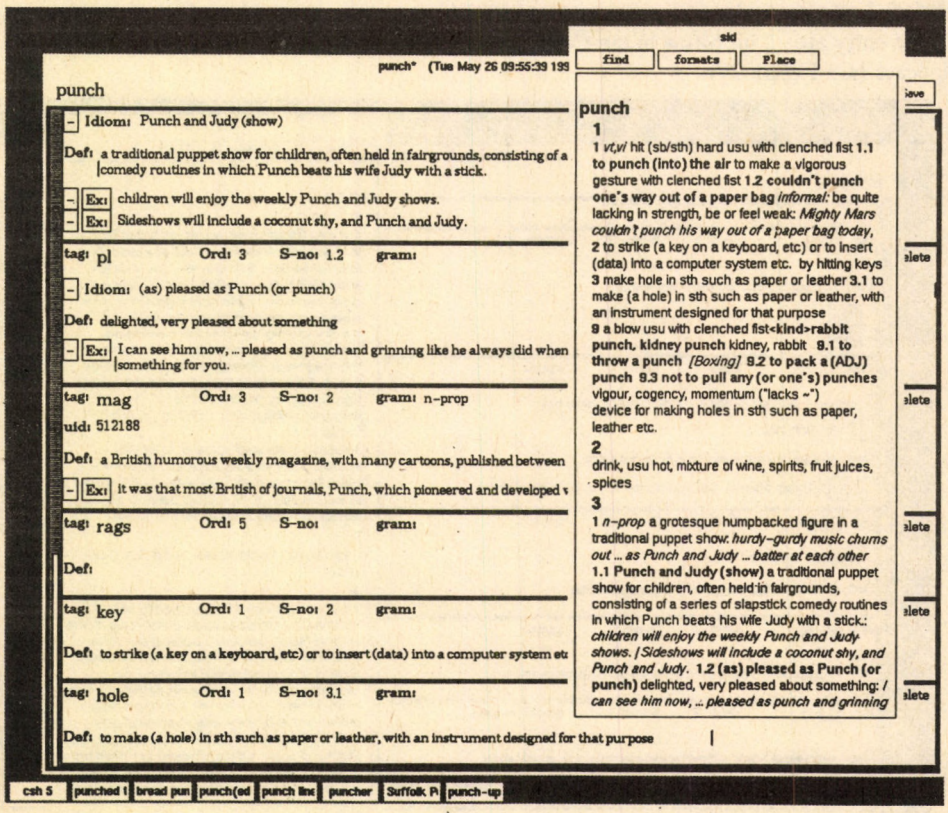


Figure 28: (Ajax) template and print versions of draft entry, verb senses numbered

The verbs outline seem acceptable, so I go back into the Ajax entry and number the noun senses, calling it up once again with **Show** to check that it is complete.

Now that I am satisfied (for the moment) with the numbering system, I reorder the senses in Ajax by hitting the **Sort Senses** button in the top command line, to make the Ajax template sense order correspond to the actual numbering in the entry. This is so that I can work down through the entry in Ajax, going back and forth into Argus to collect appropriate examples, and make sure that the facts in the Ajax entry correspond to the corpus data. Figure 29 shows part of the draft entry in Ajax format, and also a larger section from it in the print format.

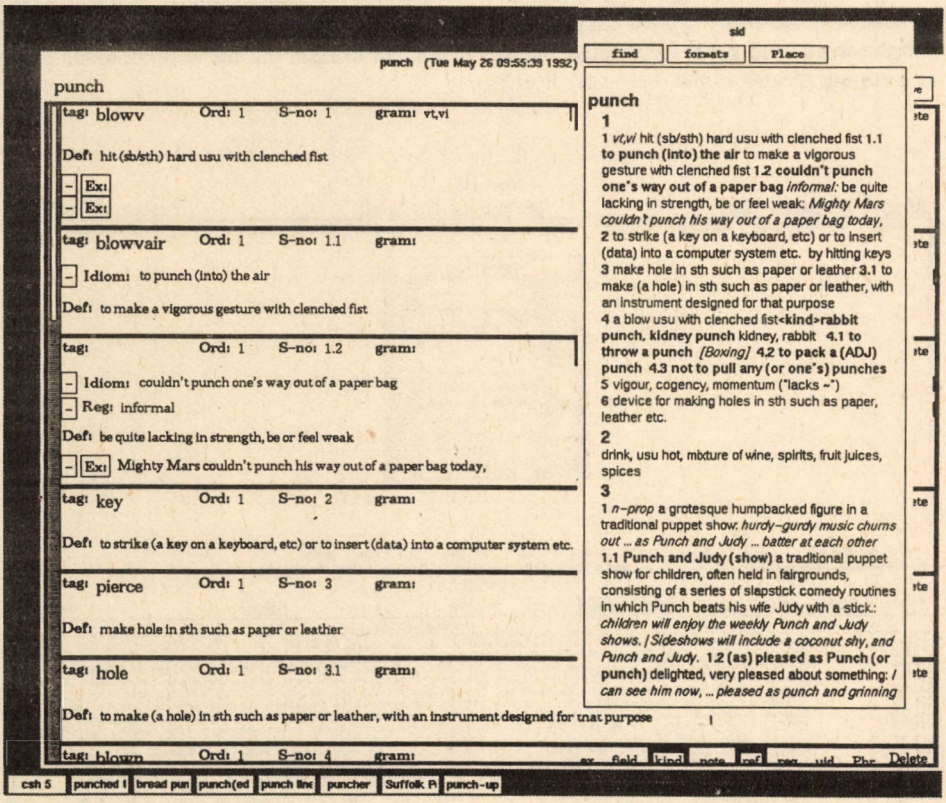


Figure 29: (Ajax) template and print versions of draft entry, verb and noun senses numbered

The work remaining to be done on fleshing out the senses in the draft entry is essentially repetitive. You work through screen after screen of material in Ajax: scan the draft definition; call up the corpus lines tagged for that sense; select typical and informative examples (for this dictionary-database, provisionally at least, length restrictions are relaxed); check for lurking idioms and collocations which have eluded you so far; polish the definition; insert the grammatical notation; insert labels relating to subject fields, register, style etc; and add any

cross-references that need to be recorded. This is a provisional and very basic dictionary entry - other material, such as pronunciations, etymologies and so on, will be added later, when the dictionary style and content are defined in more detail, omissions are repaired, and the entry is refined and polished to bring it up to standard.

I therefore start on Sense 1 ("hit sb/sth hard, usu with clenched fist"), setting up a search in Argus for all the lines tagged with the mnemonic 'blowv' (= "blow: verb") and being offered 176 concordances (see Figure 30) to scan in my search for the ideal example - one that shows the headword in a very typical context, that exemplifies constructions commonly associated with the word, and yet is informative enough in content to flesh out the definition and make the meaning clearer to the dictionary user.

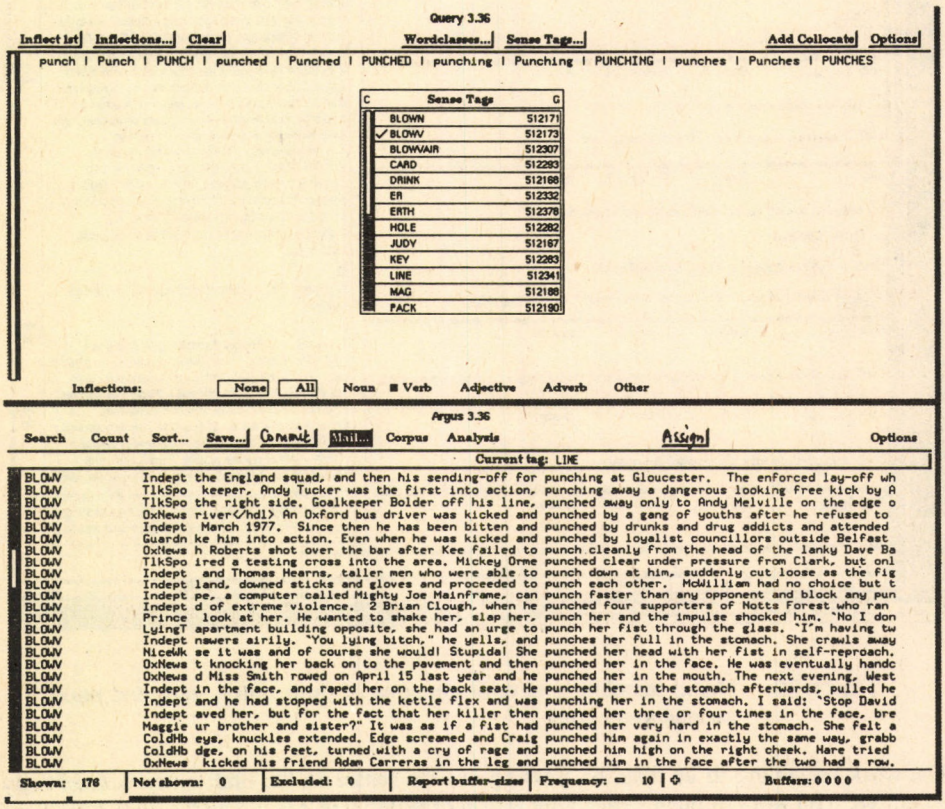


Figure 30: (Argus) citations for verb sense 1 selected on tag 'blowv'

Since *face* is one of the statistically significant collocates of *punch* (as was shown by the **coll** command, see Figure 5), and since I have noted so many lines displaying the frame "X punched Y in the BODYPART", I choose a line that exemplifies both of these features. The displayed KWIC concordances are inadequate, and I have to look at the larger context within Argus (see Figure 31), by hitting the **Corpus** button, and to cut and paste the citation from the corpus into the Ajax entry, Sense 1.

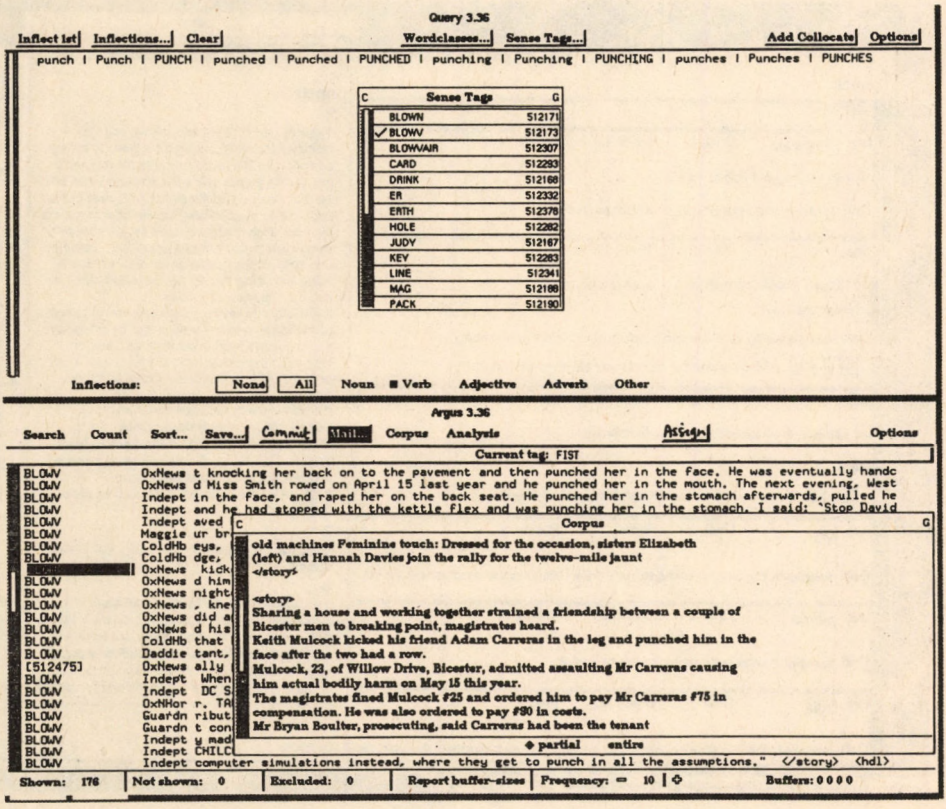


Figure 31: (Argus) expanded context for "punched him in the face"

In passing I note a couple of corpus lines with the collocation "punched his fist through .." and decide to add that as a distinct collocation in the entry. This means adding a sense frame in Ajax and extracting an example from Argus. I decide that it does not require a definition, its context in the entry making it quite transparent. However, it seems sensible to reorder the idioms within Sense 1. I change the numbers in the Ajax draft, which immediately changes the layout in the output from the Show command (see Figure 32), but it doesn't seem worth while to reorder the Ajax entry on the screen.

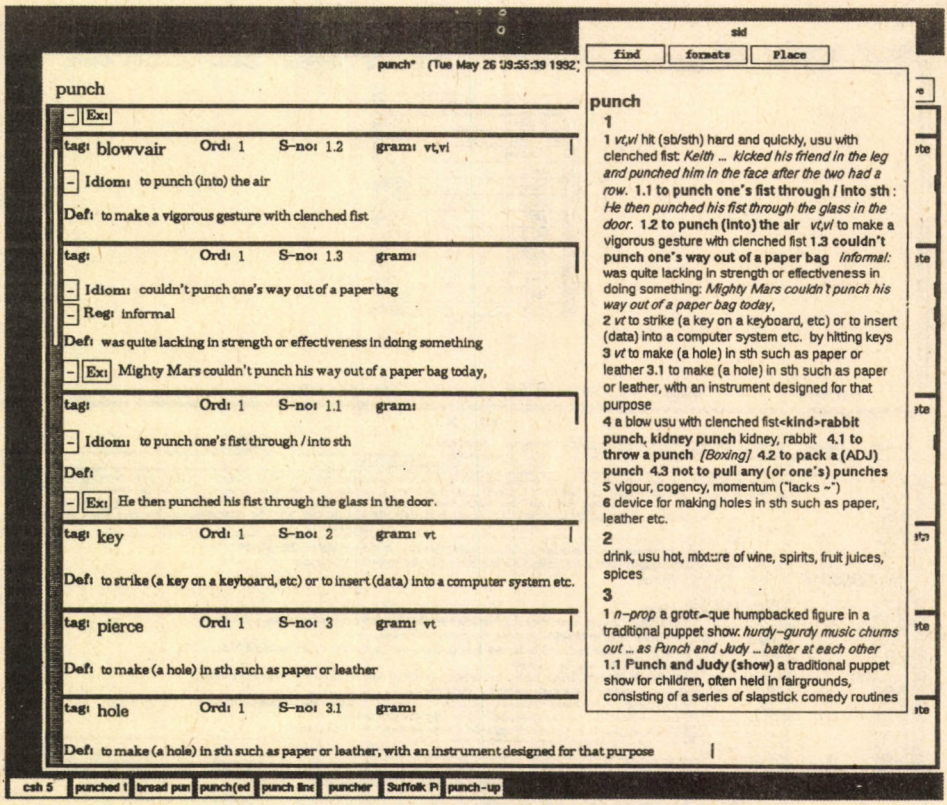


Figure 32: (Ajax) draft: template and print version from the "Show" command which automatically reorders the senses

Looking for examples of "to punch (into) the air" I change the sense-tag mnemonic in the Argus sense conditions, find one, and cut and paste it into the Ajax entry. This process continues until the verb section of the entry is complete. I pull it up with the **Show** command (see Figure 33) just to check it, before moving on to the noun.

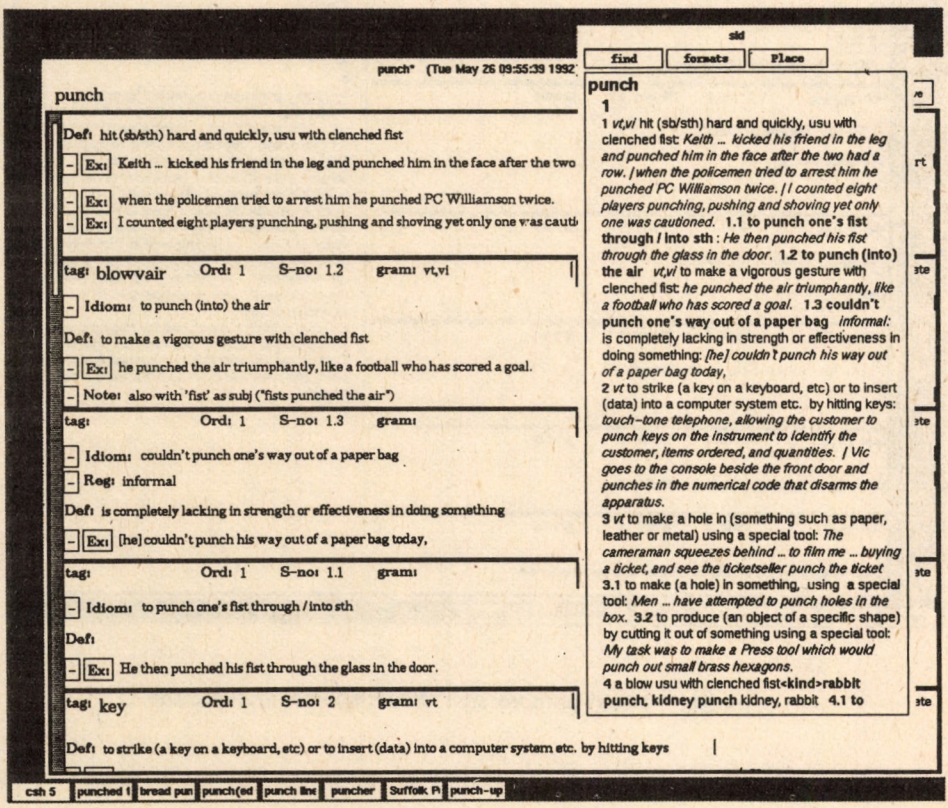


Figure 33: (Ajax) draft: template and print version with verb section complete

I study the draft noun section in the **Show** print format (see Figure 34).

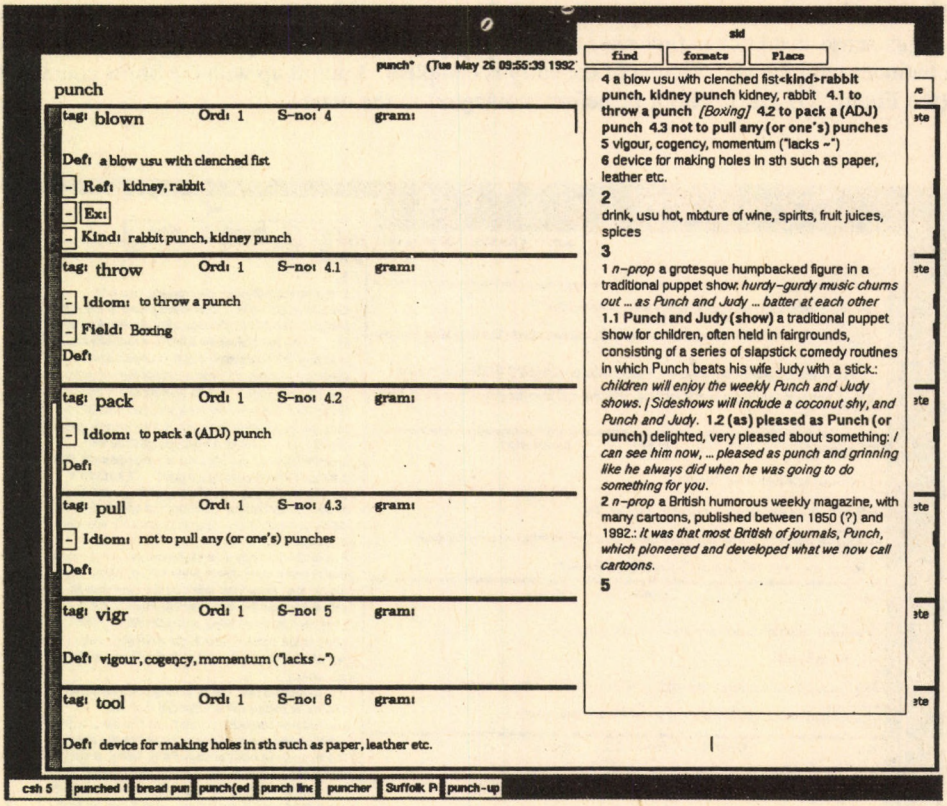


Figure 34: (Ajax) print version shows sparse noun section

This time, when I go back to the corpus for examples, I sort them on context to the left of the target word. This helps me find an example showing another significant collocate *powerful*, and also the plural use of the noun.

Finally, the entry is as complete as I wish to make it for the purpose of this demonstration, although the definitions clearly need refining, and some senses, and examples, are missing from a comprehensive description of the word. A command prints it off in some semblance of a dictionary entry (see Figure 35).

punch

punch [1]

1 *vt,vi* to hit (someone or something) hard and quickly with a clenched fist:
Keith ... kicked his friend in the leg and punched him in the face after the two had a row. || *when the policemen tried to arrest him he punched PC Williamson twice.* || *I counted eight players punching, pushing and shoving yet only one was cautioned.*
1.1 to punch one's fist through / into something
He then punched his fist through the glass in the door.
1.2 to punch (into) the air*vt,vi* to make a vigorous gesture with a clenched fist:
he punched the air triumphantly, like a football who has scored a goal.
1.3 couldn't punch his / her way out of a paper bag*informal.*is completely lacking in strength or effectiveness :
[he] couldn't punch his way out of a paper bag today.
2 *vt* to enter (numbers or other data) into a computer or other machine by striking keys on a keyboard:
Vic goes to the console beside the front door and punches in the numerical code that disarms the apparatus.
2.1 *vt* to strike (a key on a keyboard) in order to insert data into a computer or other machine:
touch-tone telephone, allowing the customer to punch keys on the instrument to identify the customer, items ordered, and quantities.
3 *vt* to make (a hole) in something, especially by making use of a special tool for the purpose:
Men ... have attempted to punch holes in the box.
3.1 *vt* to make a hole in (something such as a ticket) using a special tool:
see the ticketseller punch the ticket.
3.2 *vt* to produce (an object of a specific shape) by cutting it out of something using a special tool:
My task was to make a Press tool which would punch out small brass hexagons.
4 *nc* [Boxing, Fighting]a blow with a clenched fist:
Stanley came at me and knocked me down with a powerful punch. || *the prosecution had not proved that it was a deliberate punch and not just an accidental blow.* || *As a welterweight he proved to be less destructive; the result ... of larger men being better able to withstand the impact of his punches.*
4.1 to throw a punch[Boxing]to aim a punch at one's opponent:
further words were exchanged when Collins threw a punch after the bell which ended the first round and Laing retaliated by appearing to aim a kick at the challenger.
4.2 to swap punches*British.*to become involved in a bout of fisticuffs:
opposition MPs swapped punches and shoved over elderly members of the ruling Nationalist Party... after a disputed committee decision.
4.3 to trade punches*American.*= to swap punches
4.4 to pack a (powerful (etc.)) punchto be very effective, successful, and impressive (used of a performance, work of art, artifact etc.):
The ... Beethoven Quartet, No 16 Opus 135, packs less of a punch. || *The espresso, in small cups, is dark and serious and packs a more powerful punch.* || *Volvo's turbocharged big saloon ... packs a potent punch quite at odds with its looks.*
4.5 not pull any (or one's) punchessay something critical in a direct way, without trying to soften the impact of the words:
The young reporter pulled no punches when it came to direct questions. || *Voinovich's satire pulls no punches and hits where it hurts most.*
5 *nu* brevity and effectiveness (used of a person, performance, or the way something is told or carried out):
... communications must have clarity and cohesion. Above all they must have punch - beware of flogging an issue too hard. || *Much British cinema does lack emotional punch.*
6 *nc* a tool or machine for making holes in something such as paper, leather, metal, etc.:
showing me how the punch fitted into the guide plate

punch [2]

nu,nc a mixture of fruit juices and spices, often with wine and spirits. made usu for a party, and sometimes drunk hot:
Fruit punch was served with the meal . || *As winter drew on, many a pleasurable hour was spent in front of a blazing coal fire drinking mulled concoctions and hot punches.*

punch [3]

1 Punch*n-prop* the principal male character in a traditional puppet show, a grotesque hump-backed figure:
hurdy-gurdy music churns out ... as Punch and Judy ... batter at each other
1.1 Punch and Judy (show)a traditional puppet show for children, often held in fairgrounds, consisting of a series of slapstick comedy routines in which Punch beats his wife Judy with a stick.:
children will enjoy the weekly Punch and Judy shows. || *Sideshows will include a coconut shy, and Punch and Judy.*
1.2 (as) pleased as Punch (or punch)delighted, very pleased about something, (sometimes used to imply smugness):
I can see him now, ... pleased as punch and grinning like he always did when he was going to do something for you.
2 *n-prop* a British humorous weekly magazine, with many cartoons, published between 1850 (?) and 1992:
it was that most British of journals, Punch, which pioneered and developed what we now call cartoons.

Figure 35: Draft entry output from "printentry punch" command in Atlas

5. CONCLUSION

I believe we have all learnt - and are all learning - a great deal from this collaboration. For me, it is particularly interesting (and often chastening) to see how a new routine that seemed a stroke of genius at the drawing-board stage simply complicates the lexicographical process too much.

The task of sense-tagging all the occurrences of the headword in the corpus is proving very labour-intensive, despite the powerful tools. At present, compiling a database entry for a word with six or seven hundred occurrences, and sense-tagging these, takes a couple of days at least.

However, much effort is being put into making the programs run faster, and the interface more lexicographer-friendly. The "task list" - a wish list which the lexicographers of the team draw up, and which is systematically implemented by the computer scientists - is three pages long, and growing still. The Hector project is not finished, and the software improves and develops from week to week. This is no more than a status report, as at May 1992.

Bibliography

Atkins, B.T.S. (1987) "Semantic ID tags: Corpus Evidence for Dictionary Senses", in *The Uses of Large Text Databases*, Proceedings of the Third Annual Conference of the UW Centre for the New OED, Waterloo, Canada.

Black, E. (1988) "An Experiment in Computational Discrimination of English Word Senses", IBM Journal of Research and Development, Vol. 32, No. 2, IBM, Yorktown Heights, NY.

Byrd, R.J., Calzolari N., Chodorow M., Klavans J., Neff M., Rizk O. (1987) "Tools and Methods for Computational Lexicology", in *Computational Linguistics* 13: 219-240.

Calzolari, N. & E. Picchi (1988) "Acquisition of Semantic Information from an On-Line Dictionary", in *Proceedings of COLING 88 Budapest*, ed. D. Vargha, J. von Neumann Society for Computing Sciences, Budapest, Hungary.

Chodorow, M.S., R.J. Byrd, and G.E. Heidorn (1985) "Extracting Semantic Hierarchies from a Large On-line Dictionary", in *Proceedings of the 23rd Annual Meeting of the Association for Computational Linguistics*, 299-304.

Church, K. & P. Hanks (1990a) "Word Association Norms, Mutual Information, and Lexicography", in *Computational Linguistics* 16:1.

Church, K., W. Gale, P. Hanks & D. Hindle (1990b) "Using Statistics in Lexical Analysis", in *Lexical Acquisition: Using On-Line Resources to Build a Lexicon*, ed. U. Zernik, Lawrence Erlbaum Associates, NJ.

Church, K., W. Gale, P. Hanks, D. Hindle & R. Moon (forthcoming) "Lexical Substitutability", in *Computational Approaches to the Lexicon*, eds. B.T.S. Atkins & A. Zampolli, Oxford University Press.

Clear, J. H. (in press) "Corpus sampling" in G. Leitner (ed.) *New Dimensions in Corpus Linguistics*: Proceedings of the 11th ICAME Conference, Berlin, June 1990. Berlin: de Gruyter.

Glassman, L., C. Hibbard, J. R. Meehan, L. Guarino Reid, and M-C. van Leunen (forthcoming) "Hector: Connecting Words with Definitions."

Johansson, S. and K. Hofland (1989): *Frequency Analysis of English Vocabulary and Grammar*, Oxford University Press, Oxford.

Lesk, M. (1986) "Automatic Sense Disambiguation using Machine-Readable Dictionaries: How to tell a Pine Cone from an Ice Cream Cone", in *Proceedings of SIGDOC1*, pp. 24-26.

Levin, B. (forthcoming:) *English Verb Classes and Alternations: A Preliminary Investigation*, University of Chicago Press, Chicago, USA.

Dictionary Entry Parsing Using Standard Methods

CHRISTOPH BLÄSI — HEINZ-DETLEV KOCH

Abstract

Systems attributing structure to dictionary entries as texts are a prerequisite for several important tasks in computational lexicography; one of these tasks is the exploitation of the knowledge organized around lexemes as gathered by generations of lexicographers for natural language processing applications.
A dictionary entry parsing system is presented here which makes use of existing theoretically sound and well understood techniques and components and which can be adapted to virtually any dictionary, once a grammar for the entries in question has been set up and made accessible to the system.

1 Introduction

Dictionary entries have an implicit textual structure which is intuitively made use of by every dictionary user. There have been various attempts to make those structures explicit and to describe them formally (cf. [Wiegand 1991]). Since such descriptions are meanwhile available in a relatively explicit and formal manner, it is worth while to examine if it is possible to develop algorithms for parsing such structures. In case it is one can expect that techniques can be applied that have been developed and tested in other academic disciplines with comparable object domains.
Motivated by these considerations we have tried to refine those descriptions to such a degree that an operationalization is possible and that their correctness can be shown using a dictionary entry parser.
For three reasons, the parsing of dictionary entries has become an important field within computational lexicography :

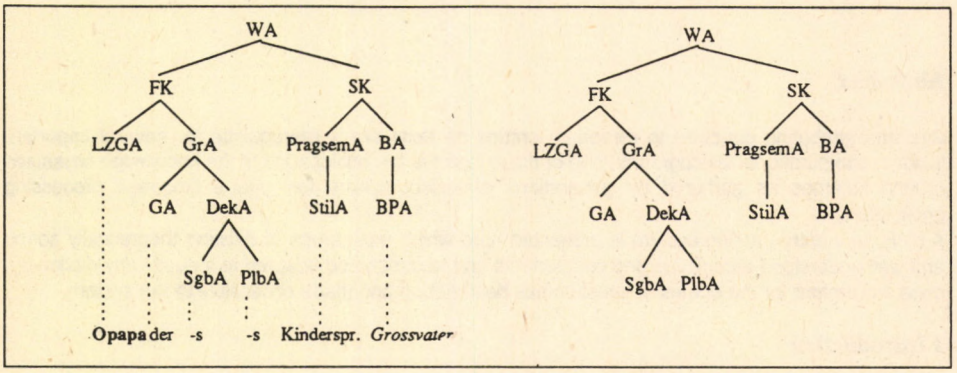
- The specific structure of dictionary entries has proven to be suitable for the representation of knowledge of language and the world. Since centuries, this knowledge has been organized around lexemes and this structure is therefore an interesting object of research.
- Since the extent of dictionaries as well as demands for their quality have increased, machine support for lexicographers is desirable also with regard to structural aspects of dictionary entries.
- From the side of natural language processing the demand for this kind of knowledge has risen to a degree which, at least for broad coverage applications, cannot entirely be satisfied by new coding.

For the qualified extraction of such knowledge the parsing of dictionary entry structures is an indispensable prerequisite (cf. [McNaught/Caroli/Hellwig 1990]).
Additional desirable features of such dictionary entry parsing systems are that they should be as general as possible (i.e, they should be applicable to dictionaries of most different kinds without substantial changes) and that they should fall back upon components of conceptionally similar applications as far as possible.
In the following, we will present a system which has been developed at the University of Heidelberg between 1990 and 1992 and its lexicographic and computational rudiments. We will demonstrate its application using examples from the DDUW (Duden Deutsches Universalwörterbuch). The system arose from several of the

listed motivations for the parsing of dictionary entries at the same time; it is especially designed to meet the requirements for generality in the sense given above and makes extensive use of already existing components.

2 Rudiments

The "meaning" of a string in a dictionary entry is determined by this string itself and its location within the dictionary entry, i.e. its position within the specific structure of the latter. The parsing of a dictionary entry can be carried out only with respect to a certain concept of structure. The choice for the constituent structure paradigm seems to be obvious in the case of dictionary entries. Immediate dominance and linear precedence relations as occurring in those entries are usually represented by corresponding constituent structure trees. With the help of the method of "exhaustive positional-functional segmentation" ([Wiegand 1989a, 1989b]) of dictionary entries such constituent structure trees can be derived. This derivation is first carried through for actual dictionary entries (see figure, left). However, abstractions can be made over whole classes of isomorphic constituent structure trees just by not considering the terminal elements (see figure, right).



For reasons of simplicity, neither typographic (e.g. font changes) nor non-typographic (e.g. comma, semicolon, colon, ..) structure indicators have been given. Abbreviations used: WA: Wörterbuchartikel / Dictionary Entry, FK: Formkommentar / Comment on Form, LZGA: Lemmazeichengestaltangabe / Item giving the Form of the Lemma Sign, GrA: Grammatikangabe / Grammar Item, GA: Genusangabe / Item giving the Gender, DekA: Deklinationsangabe / Item giving the Diclension, SgbA: Singularbildungsangabe / Item giving the Singular Formation, PlbA: Pluralbildungsangabe / Item giving the Plural Formation, SK: Semantischer Kommentar / Comment on Semantics, PragsemA. Pragmatisch-semantische Angabe / Pragmatic-Semantic Item, StilA: Stilangabe / Style Item, BA: Bedeutungsangabe / Item giving the Meaning, BPA: Bedeutungsparaphrasenangabe / Item giving the Meaning Paraphrase

For a linear representation of such tree structures production rules of the form A -> B C or list structures (A (B C)) are suitable. In the example given, A consists of B and C or B and C are parts of A, respectively. The immediate dominance and linear precedence relations stated by these rules are the by far dominating ones, but not - as hinted at above - the only relations occuring in dictionary entries; for the representation of further relations (e.g. the addressing relation and the scope relation, which are both independent from the constituent structure) a representation as tree structure or production rule does not suffice. For this, one needs corresponding annotations the details of which are beyond the scope of this paper (cf. [Bläsi 1991a, 1991b, 1991c]). Actual dictionary entries can be attributed structure by mapping them to the abstractions of the corresponding class of dictionary entries. The rules generating the set of all such abstract constituent structure trees are taken as grammar. A slightly simplified extract from the grammar for noun entries of the DDUW is given below:

```
WA    -> FK %EKA EtyA %EKZ SK %PKT
WA    -> FK SK
FK    -> bold LZGA %KOM normal GrA
GrA   -> GA %SEM DekA
GA    -> TEXT
DekA  -> SgbA %KOM PlbA
```

%EKA: Angular bracket, opening; %EKZ: Angular bracket, closing; %PKT: Period; %KOM: Comma; %SEM: Semicolon

3 The Analysis Technique

The structural analysis is carried out relative to an extended version of the phrase structure grammars introduced in section 2. Techniques for the analysis of strings based on these context-free grammars have been examined and compared for formal as well as for natural languages ([Aho 1987], [Hellwig 1990], [Tremblay 1985]). They are applied to a large extent in the various disciplines of computer science as well as in computational linguistics. For our objective we have chosen a special technique geared to the analysis of formal languages. Below, we will describe this technique and the modifications made for the processing of natural language and, in an additional step, dictionary entries.

The chosen technique has originally been developed for a subset of formal languages. This set of languages can be described by certain context-free grammars which are called LR(1) grammars. An operational definition of this class of languages is based on the definition of an abstract, deterministic, finite automaton, which is seen as a representative of a LR(1) grammar. All languages which can be generated by such an automaton belong to the class of the so-called LR(1) languages which also includes most of the programming languages used today. LR(1) languages have the advantage that the abstract automaton which generates such a language can likewise be applied to its analysis. Fortunately, this analysis can be performed in linear time.

Genuine LR(1)-languages are built on atomic categories and do not allow for ambiguous grammars. When analysing an input string, the automaton steps through a finite set of states and, triggered by its current state and the current input symbol, decides whether to push the current input symbol on a stack (Shift) or to replace some of the symbols already on the stack by a nonterminal symbol (Reduce). By performing any of these actions, the automaton changes its state according to the symbol on top of the stack and the current input symbol. A grammar is an LR(1)-Grammar if one could construct an automaton that contains only one possible action for every pair of a state and a symbol.

LR(1)-Grammars were modified for natural language processing by Tomita ([Tomita 1985a, 1985b]) who introduced a graph-structured stack to allow for multiple alternative actions for any state-symbol combination. If the automaton encounters a state with more than one possible action the algorithm forks the stack and pursues the resulting alternatives in parallel. Analysis of an input string succeeds if at least one of the alternatives succeeds.

We have further modified the Tomita approach by allowing for complex, i.e. feature augmented, categories which contain pairs of features and feature-values. By imposing restrictions on the possible feature values, we can bidirectionally enforce identity of values by mutual instantiation of variables or restrict values to sets of admissible values for a feature. By restricting unification processes to local trees, i.e. trees of depht 1, we are avoiding the time-consuming complexity of full graph unification and are thus combining the speed of LR(1)-analysis with the expressive power of general context-free grammars.

As input for the parsing process we are using the machine-readable representation of the dictionary, i.e. the computer typesetting tape.

4 The Processing of the Typesetting Tape

A typesetting tape is the result of a computing operation geared to a specific typesetting machine which processes the input text according to the typographic requests on the part of the typesetter. Among other tasks, this operation determines the distribution of the textual material to lines, colums, pages and sheets. As the result of this step, a typesetting tape contains the informational content as well as informations concerning the typography and layout of the text. The latter informations are coded in a language specific to the typesetting machine used. Some of them, as for example the ones concerning the page makeup, are totally irrelevant for determining the structure of a dictionary entry. Others, however, especially font changes, provide indispensable information on the organisation of the dictionary entry. Without the latter an entry is hardly comprehensible even to the human reader of the printed version.

Whereas irrelevant control sequences as exactly e.g. the ones concerning the page makeup or additional blanks after italicised text passages can thus safely be ignored, other information carrying control sequences have to be recognized. They can, moreover, be replaced by less machine dependent and more generally understandable characters. Before the processing, an extract of the type setting tape of the DDUW looks like this:

```
ð5û¨-Ona+ger, ð1ûder; -s, - ¦lat. onager, onagrus }} }¦}
griech. 'onagros; 2: nach der einem Esel glei}-}
chenden Form¦: ð3û1. ð2ûin Südwestasien heimi}-}
scher Halbesel.ðVR10û ð3û2. ð2û(im antiken Rom) Wurfma}-}
schine.æ
```

This typesetting tape produces the following print output:

Qnaiger, der; -s, - [lat. onager, onagrus < griech. ónagros; 2: nach der einem Esel gleichenden Form]: 1. *in Südwestasien heimischer Halbesel.* 2. *(im antiken Rom) Wurfmaschine.*

In a first step of the processing of the typesetting tape, control characters specific to the typesetting system are replaced by the ones chosen for the analysis process, structurally irrelevant sequences are deleted. After this step the same extract of a typesetting tape looks like this:

```
%FONT5 %USTR1 Ona %VSTR ger %KOM %FONT1 der %SEM %HSTR s %KOM %HSTR
%EKA lat %PKT onager %KOM onagrus %LT griech %PKT %ALR onagros %SEM 2
%DPP nach der einem Esel gleichenden Form %EKZ %DPP %FONT3 1 %PKT
%FONT2 in Südwestasien heimischer Halbesel %PKT %FONT3 2 %PKT %FONT2
%KLA im antiken Rom %KLZ Wurfmaschine %PKT
```

The replacements mentioned above are of a purely textual nature and can be carried through with any text editor.
Owing to the multifunctionality of the remaining delimiters the "textual" units can not yet be identified as string units of an entry in a straight-forward way. Those delimiters have a structuring function as punctuation marks on the linguistic level on the one hand and explicate the organisation of the dictionary entry as condensed text on the other. This class of structure indicating information cannot be separated from the informational content without some effort. Details will be given in section 5.

5 The Segmentation Problem

Since the system in question is a dictionary entry parser, its structural view is necessarily restricted to structures constituting the dictionary entry as the object of investigation. Every item (as a string at a certain position in the dictionary entry structure), however, has itself a (subordinated) structure (namely e.g. a sentence structure or a phrase structure - this can, of course, be continued to the morphological and phonological level). Those subordinated structural levels are not accessible to the view of a dictionary entry parser. The problem is that it is not clear from the outset which sentences and phrases, respectively, are units not to be pervaded further by the dictionary entry parser and which are not.
We have tried several strategies to solve this segmentation problem. Only one of those solutions, the most radical and least "intelligent" one, succeeded.

- All strings which appear between certain types of structure indicators (semicolon, colon, font change, ...) are interpreted as string units.From what has been said in the previous paragraph it is clear that this approach must fail. Some of the delimiters can mark the boundary of string units as well as occurr within such units. An angular bracket in the DDUW can e.g. mark the boundary of an Item giving the Etymology (as string unit, from the Comment on Form and from the Comment on Semantics) as well as introduce the giving of facultative letters within a Grammatical Item.
- Everything which lies between certain boundaries is interpreted as a string unit. These boundaries have to be found - with the aid of the structure indicators already mentioned - by context-sensitive matching (b) or with the help of a context-sensitive transition network. As for angular brackets it has to be decided, if they are situated within a Grammatical Item (in this case the angular bracket just occurrs within a string unit) or not (the angular bracket marks the boundary between string units). This approach is inappropriate because much of the structural knowledge which desired as the result of the actual parsing process is needed for the preprocessing already. For the decision on the angular bracket one needs to know if an Item giving the Form of the Lemma Sign has been consumed or not.

- A "maximal" segmentation is performed. All typographic structure indicators and all delimiters which in at least one of their usages can mark such boundaries are taken to be the set, with respect to which this segmentation is carried through. Strings which have been split "overeagerly" are re-concatenated in a later stage of work - namely during parsing. This applies especially to the case that a delimiter has been interpreted as marking a boundary, whereas it functioned as a punctuation mark. For this approach, however, the grammar has to be slightly changed. It has to be allowed that items can be realized not only as units, but alternatively as chain of such units. If one makes sure that the corresponding rules are defined strictly right or strictly left recursively this strategy leads to very satisfying results. (The latter precaution is to prevent that the same overall string occurrs several times because of various derivation possibilities from smaller component units; the runtime efficiency would be influenced in a very negative way.)

6 Description of the Implementation

The grammar for the analysis is written in a formalism following the programming language Lisp and resembles the usual notation for production rules. A special preprocessing program compiles this grammar into a control table for the automaton. The output of the preprocessing step is loaded by the actual analysis program which contains a rudimentary LR(1)-automaton. By using this control table the analyzer is capable of attributing structure to all input strings which are covered by the grammar.
An extract of the grammar for the DDUW which describes the arrangement of the Comments on Semantics can be represented in production rule notation as follows:

```
SK    -> %DPP  SKK
SKK   -> PA SSK
SKK   -> PA SSK %PKT SKK
```

This fragment dwells on the fact that a Comment on Semantics consists of the categories *%DPP* (for the non-typographic structure indicator colon) and *SKK* (for the Complex of Comments on Semantics), whereas *SKK* itself consists of a *PA* (Item giving Polysemy) and a *SKK* (Subcomment on Semantics). The third rule determines that a Complex of Comments on Semantics can be followed by further Complexes of Comments on Semantics which are seperated by *%PKT* (for period). The formalism as used by the system differs from this notation syntactically by just combining all involved categories in a list the first element of which is the superordinate unit. The fragment shown above is rewritten as follows:

```
[1]   ( (SK) (%DPP) (SKK) )
[2]   ( (SKK) (PA) (SSK) (%PKT) (SKK) )
[3]   ( (SKK) (PA) (SSK) )
```

The necessity for additional bracketing of every category is obvious if we consider a rule which shows complex categories. In this case all specifications related to a category are combined in the one bracket, whereas the features and the feature values are themselves enclosed in brackets. Feature values which have to be instantiated identically are notated as bracketed variables. Variables with identical names have to be instantiated by identical values at runtime if they occurr in the same production. The rule below describes the form of an Item giving Polysemy in the DDUW:

```
[4]   ( (PA (index (x))) (TYPO-ANF (art fett)) (TEXT (string (x)))
(%PKT) )
```

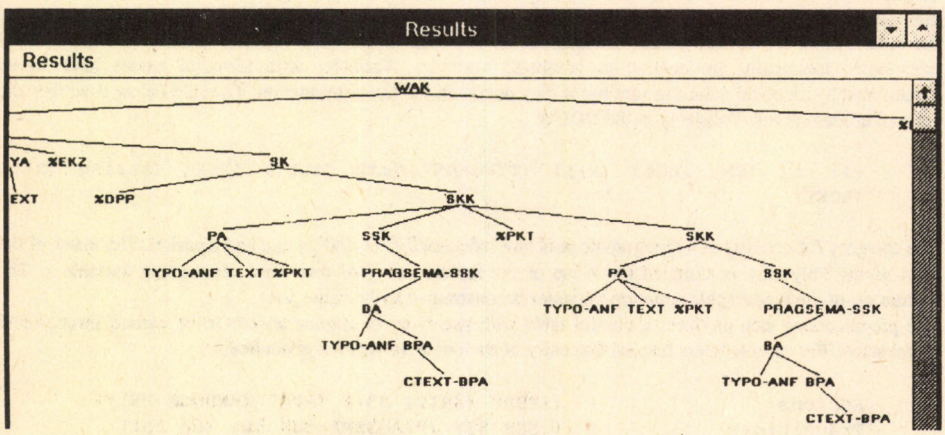
The category *PA* consists of a typographic structure indicator *TYPO-ANF*, a text and a period. The index of the Item giving Polysemy is assigned the value of the feature *string* of *text* via a reoccurring variable *x*. The feature *art* of the typographic structure indicator is constrained to the value *fett*.
The preprocessing step produces a control table with two types of entries: actions to be carried through and target states. For completeness reasons one entry of each of these types is given below:

```
Actions:                ((%DPP (SHIFT 33)) (%PKT (REDUCE 28)))
Transitions:            ((SSK 57) (PRAGSEMA-SSK 56) (BA 55))
```

The program takes about 3.5 minutes to produce a control table with 224 states for a grammar with 117 rules. Since the control table is written to a file which is loaded by the actual analysis program, this step has to be performed only when the grammar has been changed.
The analysis program Paula (Parser for Ambiguous Unification grammars / Lr(1)-Analysis) is completely written in Lisp and contains essentially a rudimentary LR(1) automaton, a component for the lexicon search and a module for graphic output of the results. Apart from the control table and the actual input, it requires a lexicon which establishes the connection between the units occurring in the input text and the categories used for the grammar. The units occurring in the lexicon are the ones in the processed typesetting tape. An extract of the lexicon used for the analysis of the DDUW has the following form:

```
(%KLA  %KLA (string "(") )
(%KLZ  %KLZ (string ")") )
(%SEM  %SEM (string ";") )
(%PKT  %PKT (string ".") )
(%DPP  %DPP (STRING ":") )
(%KOM %KOM (string ",") )
(%SKA  %SKA (string "<") )
(%SKZ  %SKZ (string ">") )
(%FONT5 TYPO-ANF   (ART fett))
(%FONT1 TYPO-ANF   (ART normal))
(%FONT2 TYPO-ANF   (ART kursiv))
```

Again, the notation is oriented towards Lisp and describes a lexicon entry as a list. The first element of the list is the symbol actually occurring in the processed typesetting tape, whereas the rest of the list contains the information to be provided by the lexicon component, as soon as this unit occurrs in the input. For the analysis of dictionary entries, the lexicon component of the parser was modified so that it provides the category *text* if a string enclosed in double quotes occurrs in the input. This string is attributed to the feature *string* of this category. Therefore all elements of the input which have been enclosed in double quotes by the typesetting tape processing program can be treated by the parser as string units. In the rule (4) above the index of an Item giving Polysemy, i.e. the number in front of the period taken as text, is attributed to the feature *string* of the category *text* by the lexicon phase of the parser and thus propagated to the superordinate category *PA*.
Initially, the program loads the control table as well as the lexicon and starts with the analysis of the input file which may contains an arbitrary number of articles from the preprocessed type setting tape. After the analysis of an entry, its structure is usually displayed in a window which the user can move over the constituent structure tree. When choosing a category with the mouse, its accompanying features are displayed in an additional window. The figure below shows an extract of the structure of the DDUW entry ONAGER as displayed by the output component:



Alternatively, the structures produced can be written to an output file in list notation. This output can be processed further by other programs. Such programs could be used to extract certain types of information, e.g. grammatical information, or to transfer the results to a lexicographic database (cf. [Koch 1992]). Optionally, all not analyzed inputs, i.e. all entries which were not covered by the grammar, can be written to a second output file to be postprocessed manually or to serve as guide for the extension of the grammar. Paula puts two tools for the grammar development at the disposal of the user. It is possible to run the analysis in single step mode. The program interrupts the analysis after each action and offers the possibility to examine the actual status of the automaton as well as to check the analysis steps carried out already. As a further option, Paula can interrupt the analysis if one of the running analysis processes terminates without a result in order to give the program user the possibility to investigate the reason for the failure. The usual proceeding for drawing up a grammar consists in developing a rather small grammar for a part of the entries to be examined which is applied nonetheless to the entire typesetting tape. Entries not analyzed are written to a file. By examining those not accepted it is usually possible to determine rather soon where and in which way the grammar has to be extended. In case of a lack of clarity concerning the reason for the failure, the mechanisms described above can be applied to gain more detailled information. Using the modified grammar for the next analysis attempt the number of not accepted entries is supposed to decrease and allows for a renewed extension of the grammar.
The output component has to cater for the consequences of the segmentation strategy chosen (cf. section 5). The segmentation of the typesetting tape into units as small as possible results in inadequately hierarchical subtrees consisting of recursively descending pairs of delimiters and rest trees for the textual constituents. The subordination of these constituents under the same textual category, however, indicates the fact that during the processing of the typesetting tape a too finegrained segmentation has been assumed. Therefore the produced constituent structure tree is searched for the occurrence of textual constituents after the actual analysis process has finished. All constituents directly or indirectly subordinated to such a category are deleted and the values of their string feature are "collected" and concatenated. The string resulting from this is attributed to the feature compound-string of a new main node, replacing the whole subtree.

Technical details of the implementation

The program producing the table as well as the analysis program are totally independent on the dictionary to be examined. Paula takes about 1.5 seconds for the analysis of a dictionary entry of average size and complexity using the grammar applied at the moment. For the implementation we used Golden Common Lisp, Version 4.1 with the Gold Hill Windows extension, Version 4.1 under Microsoft Windows 3.1 on a IBM compatible PC with a 33 Mhz 80386 processor which is equipped with 9 MB RAM and a SVGA graphics card with 800 X 600 pixels.
For the processing of the typesetting tape we used an ordinary text editor with macro facility (KEdit 4.0) and a simple C program. These editor macros and the C program, however, are dependent of the typesetting system used and have to be adapted accordingly if the system is applied to a dictionary which is produced by a different typesetting system or read with an OCR scanner.
Independently from this fact, a grammar has to be drawn up for each dictionary to be examined.

7 Example

In the following, the structure produced for the DDUW entry ONAGER is shown. To improve readability, the lists have been indented according to their structure and informations not relevant for the entry structure have been removed . The markings :T, :C, and :F occurring in the output indicate the type, the constituents, and the features of the different categories.

```
(:T WAK  :C
     ((:T WA  :C
         ((:T FK :F ((KLASSE SUBST)) :C
              ((:T TYPO-ANF :F ((ART FETT)))
              (:T LZGA  :C
                   ((:T CTEXT-LEM :F ((COMPOUND-STRING "__Ona!!ger")))))
              (:T %KOM :F ((STRING ",")))
              (:T TYPO-ANF :F ((ART NORMAL)))
```

```
        (:T GRA  :C
            ((:T GA :C
                ((:T TEXT :F ((STRING "der")))))
             (:T %SEM :F ((STRING ";")))
             (:T DEKA :C
                ((:T SGBA :C
                     ((:T %HSTR :F ((STRING "-")))
                      (:T TEXT  :F ((STRING "s")))))
                 (:T %KOM :F ((STRING ",")))
                 (:T PLBA  :C
                     ((:T %HSTR :F ((STRING "-")))))))))))
    (:T %EKA :F ((STRING "[")))
    (:T ETYA  :C
        ((:T CTEXT :F ((COMPOUND-STRING "lat.onager,onagrus<griech.
                                         'onagros;2:nach der einem Esel
                                         gleichenden Form")))))
    (:T %EKZ :F ((STRING "]")))
    (:T SK  :C
        ((:T %DPP :F ((STRING ":")))
        (:T SKK  :C
            ((:T PA :F ((NUMMER  "1")) :C
                ((:T TYPO-ANF :F ((ART FETT) (SUBTYP ?)))
                 (:T TEXT :F ((STRING "1")))
                 (:T %PKT :F ((STRING ".")))))
             (:T SSK  :C
                ((:T PRAGSEMA-SSK  :C
                     ((:T BA  :C
                         ((:T TYPO-ANF :F ((ART KURSIV)))
                          (:T BPA :C
                              ((:T CTEXT-BPA :F ((COMPOUND-STRING
                        "in Südwestasien heimischer Halbesel")))))))))))
             (:T %PKT :F ((STRING ".")))
             (:T SKK  :C
                ((:T PA :F ((NUMMER "2")) :C
                     ((:T TYPO-ANF :F ((ART FETT) (SUBTYP ?)))
                      (:T TEXT :F ((STRING "2")))
                      (:T %PKT :F ((STRING ".")))))
                 (:T SSK  :C
                     ((:T PRAGSEMA-SSK :C
                         ((:T BA :C
                             ((:T TYPO-ANF :F ((ART KURSIV)))
                              (:T BPA :C
                                  ((:T CTEXT-BPA :F ((COMPOUND-STRING
                      "(im antiken Rom)Wurfmaschine")))))))))))))))
(:T %PKT :F ((STRING ".")))))))
```

8 Related Work

There have been several approaches to dictionary entry parsing. The approach followed by IBM (cf. e.g. [Neff/Boguraev 1989]) is probably the most widely known among them. Continental applications of the underlying ideas have been described for example by [Wermke/Bläser 1990] and by [Marinai/Peters/Picchi 1990]. As far as we can see, these approaches have analyzed a wide range of interesting and complex phenomena in dictionary entries. Moreover, they have reached an astonishing high degree of parsing coverage. However, we have shifted emphasis in at least two dimensions:

- The rules used in our system are as it were natural by-products of actions in the established lexicographic theory on dictionary microstructures ([Wiegand 1989a, 1989b]). The writing of a corresponding dictionary entry grammar falls directly back to the results of the aforementioned "exhaustive positional-fuctional segmentation" (and, we would like to add, classification) in this theory. This segmentation method deserves the name method in so far as all segmentation steps are described algorithmically. Moreover, a clear terminology for these well-defined segments has been introduced which further eases the handling of the rules.

As opposed to all this, writing rules in the IBM paradigm includes necessarily ad-hoc-theorising on the nature of dictionary entry structures and ad-hoc-naming of certain parts of them.

- We consider dictionary entries as texts with a characteristic text structure which therefore should be made explicit without destroying its effects on the appearance in printed form. Dictionary entries are the result of a condensation process from running text describing linguistic entities. It is idiosyncratic to single

dictionaries, constitutes their styles to a large degree and cannot easily be formulated as algorithms. Using attribute-value-pairs for the indication of "cross-tree" relations (e.g. scope and addressing as mentioned above), we can make all relations, including these cross-tree ones, explicit while preserving the structure of the dictionary entry structure as a condensed text structure.

Whereas the strategy "to represent precisely all information contained in the printed entry - and, wherever possible, to represent explicitely information which is only given implicitly" ([Marinai/Peters/Picchi 1990]) means duplicating subtrees with information referring to (e.g.) two other subtrees in a non-trivial way, it is our strategy to uniquely mark this subtree as refering to the two others, i.e. as having those two in its scope.

9 Further Perspectives

As for the second and the third of the possible motivations for dictionary entry parsing in computational lexicography given in the introduction, this system certainly proves the feasibility of the approach described above. By developing a running dictionary entry parsing system, the information contained in printed dictionaries (or the accompanying typesetting tapes, respectively) is by far not exploited and processed exhaustively.

The following steps are conceivable and desirable for the future:

- The informational content of the units not to be considered with regard to their internal structure in the present context, for example the Items on Meaning, should be treated as such and represented in a suitable manner. This has e.g. been done in the ACQUILEX project (cf. [Calzolari/Zampolli 1989]).
- Since there is at most an arbitrary and traditional relation between the questions of a potential dictionary user and the place in the dictionary entry in which this particular question is answered, different syntactic configurations of the same "answer" across different dictionaries should be given identical semantics. Therefore, dictionary structures as wholes should be examined with respect to their semantics. Even if dictionary entries cannot be interpreted as functions or other well-researched mathematical objects, there is always the possibility for translation semantics which just defines the semantics of the "language" in question in terms of another "language" or formalism which has been given semantics already. Representatives of such semantically well-defined formalisms - some of which in addition have the advantage of being familiar to most of the lexicography community - are PROLOG, DATR (cf [Gazdar/Evans 1990], [Gibbon/Ahoua 1991]) and Typed Feature Structures ([Ait-Kaci 1986]). Owing to the nature of lexical knowledge, default and inheritance mechanisms are needed for reasons of generalisation.

The fact that OPAPA has an "s" as its plural ending (cf. section 2) is "coded" along the path LZGA - FK - GrA - DekA - PlbA in the DDUW. A desirable translation into (Pseudo-) PROLOG would be

```
Has_Plural_Ending (Opapa, s)
```

one in DATR

```
Opapa: <FK LZGA>                 ==    Opapa
       <FK GrA DekA PlbA>        ==    s
       <>                        ==    Noun_rule.
```

Necessarily, the predicates (in the PROLOG case) and path elements (in the DATR case) should be constrained to a defined repertoire oriented to a taxonomic level of linguistic description ([Hellwig/Minkwitz/Koch 1990], cf. [Heid/McNaught 1991]) and/or potential dictionary user questions (corresponding studies are a desideratum).

With the help of "resolutions" of the type

```
Has_Plural_Ending (X, s) -> Has_Plural (X, X^s) ("^" is the concatenation operator)
```

or

```
Noun_Rule:    <Plur>=="<FK LZGA>" ^ "<FK GrA DekA PlbA>". ("^" as above)
```

respectively, lexical generalisations can be tackled appropriately.

Technically, such translations (yielding semantics implicitly via the aforementioned translation semantics) can rather easily be achieved during parsing by means of a syntax oriented translation using attributed grammars ([Knuth 1968], cf. [Deransart/Jourdan/Lorho 1988]).

The translation results are dictionary-independent and it is thus possible to load different source dictionaries with different dictionary entry structures into a common lexical database: additional information can be ignored if stored already and added if not. In the case of conflicting information, strategies as for example "Dictionary A is usually more reliable than dictionary B" can be applied.

For both steps a running, theoretically sound and well understood dictionary entry parsing system is necessary.

Finally we would like to thank Duden Publishers, especially Dr. Matthias Wermke, for the kind support of our work.

References

AHO, A.; SETHI, R.;ULLMAN, J.: Compilers, (1987)

AIT-KACI, H.: An Algebraic Semantics Approach to the effective Resolution of Type Equations, in: Theoretical Computer Science 45 (1986), pp. 293-351.

BLÄSER, B., WERMKE, M.: Projekt "Elektronische Wörterbücher/Lexika": Abschlußbericht der Definitionsphase, IWBS [Institut für wissensbasierte Systeme der IBM Deutschland] Report 145, Heidelberg (November 1990)

BLÄSI, C.: Einige Überlegungen zum WA-Parsing ..., Internal Paper, Heidelberg (January 1991)

BLÄSI, C.: Einige weitere Überlegungen zum WA-Parsing ..., Internal Paper, Heidelberg (February 1991)

BLÄSI, C.: A forthcoming system for the "reusable" parsing of dictionary entries, Internal Paper, Pisa (May 1991)

BLÄSI, C.; KOCH, H.-D.: Maschinelle Strukturerschließung von Wörterbuchartikeln mit Standardmethoden, to appear in: Lexicographica 7.1991 (1992)

CALZOLARI, N.; ZAMPOLLI, A.: Lexical Databases and Textual Corpora: a trend of convergence between Computational Linguistics and Literary and Linguistic Computing, in: Proceedings of the ALLC/ACH Conference in Toronto 1989

DERANSART, P.; JOURDAN, M.; LORHO, B.: Attribute Grammars, Heidelberg (1988)

DROSDOWSKI, G. et al. (ed.): Duden Deutsches Universalwörterbuch, Mannheim/Wien/Zürich (^{2}1989)

EVANS, R; GAZDAR, G. (eds): The DATR papers. Cognitive science research paper 139. University of Sussex, (February, 1990).

GIBBON, D.; AHOUA, F.: DDATR: un logiciel de traitement d'heritage par default pour la modélisation lexicale, English/Linguistics Interim Report No. 4, Bielefeld (May 1991)

HEID, U., MCNAUGHT, J.: Eurotra-7 Study: Final Report, Bruxelles/Stuttgart (August 1991)

HELLWIG, P.: 31 - Parsing natürlicher Sprachen: Grundlagen, in: Computational Linguistics Computerlinguistik (HSK 4), Berlin/New York (1990)

HELLWIG, P.: 32 - Parsing natürlicher Sprachen: Realisierungen, in: Computational Linguistics Computerlinguistik (HSK 4), Berlin/New York (1990)

HELLWIG, P; MINKWITZ, T.; KOCH, H.-D.: Eurotra-7 Study, DOC-9: Standards for Syntactic Description, Bruxelles (1991)

KNUTH, D.E.: Semantics of Context-free Languages, in: Mathematical Systems Theory 2,2 (June 1968), pp. 127-145. Correction: Mathematical Systems Theory 2,5, (March 1971), pp. 95-96

KOCH, H.-D.: Eine Entwicklungsumgebung für Lexikalische Datenbanken auf Basis typisierter Merkmalsstrukturen, Internal Paper (forthcoming), Heidelberg (1992)

MARINAI, E., PETERS, C., PICCHI, E.: The Pisa Multi-Lexical Database System, An integrated system for the acquisition, maintenance and interrogation of mono- and bilingual LDBs, ACQUILEX, ESPRIT BASIC RESEARCH ACTION No. 3030, Twelve Month Deliverable, Pisa (November 1990)

MCNAUGHT, J., CAROLI, F., HELLWIG, P.: Eurotra-7 Study, DOC-1: Possible Applications of Reusable Lexical Resources, Bruxelles (October 1990)

NEFF, M.; BOGURAEV, B.: Dictionaries, Dictionary Grammars and Dictionary Entry Parsing, in: 27th Annual Meeting of the Association for Computational Linguistics, Proceeding of the Conference, Vancouver (1989)

TOMITA, M.: An efficient context-free parsing algorithm for natural language, in: Proceedings of IJCAI 1985

TOMITA, M.: Efficient parsing for natural language, Dordrecht (1985)

TREMBLAY, J.-P.; SORENSON, P.: The Theory and Practice of Compiler Writing, New York (1985)

WIEGAND, H. E.: 38a - Der Begriff der Mikrostruktur: Geschichte, Probleme, Perspektiven, in: Wörterbücher Dictionaries Dictionnaires (HSK 5.1), Berlin/New York (1989)

WIEGAND, H. E.: 39 - Arten von Mikrostrukturen im allgemeinen einsprachigen Wörterbuch, in: Wörterbücher Dictionaries Dictionnaires (HSK 5.1), Berlin/New York (1989)

WIEGAND, H. E.: Printed dictionaries and their parts as text, in: Lexicographica 6.1990 (1991), pp. 1-126

Text based dictionary work for a domain-specific language

ANNA BRAASCH

The need for detailed, consistent and elaborated dictionaries covering different domain-specific languages is common to human translators and machine translation systems. This paper deals with some requirements to the lexical description for use in machine translation (MT). The subject domain is motor/car mechanics and the topic is described in the text type 'Owners Manual'.

The main point is to elaborate a method for description of lexical units including both linguistic and conceptual knowledge. The represented information types can be used in sublanguage dictionaries both for human and for machine translation purposes.

Project description
A broadly defined project on translation of technical literature was initiated by The Danish Research Council for the Humanities in 1990. The project consists of five sub-projects, its overall objective is to collect experience and knowledge from different fields such as dictionary use in the translation process, linguistic and domain specific knowledge needed for high-quality translation, and translation strategies for humans and machines. One of the sub-projects is carried out at the Center for Language Technology (CST).

Sub-project 3: Machine translation aspects
CST, the successor of the Danish EUROTRA group, is engaged in research, development and implementation in the field of computational linguistics: one of its relevant tasks is machine translation, as a follow-up of the MT-project of the EC. The work in this area also includes e.g. feasibility studies for utilization of the existing translation system outside the EUROTRA-project.

Basis and background for the described project
The aim of this contribution is to present a project concerning the application of the *Eurotra Translation System (ETS)* to a particular text type and sublanguage area.

MT systems normally contain two main rule modules: the grammar and the lexicon

components. The quality of a translation depends very much on the content, structure and granularity of the information types represented in the lexicon. The basic requirements for descriptions in MT-lexicons are well known: explicitness, exhaustiveness, consistency, unambiguousness and formalization.
The lexicon and grammar components originally elaborated for the ETS did not cover the newly chosen domain and text type: motor/car mechanics, presented in the text type 'Owners Manual'. This particular domain and the text type were chosen in cooperation with the other sub-projects, but on the specific criteria of suitability for MT.
In discussing knowledge representation for natural language processing (NLP) purposes - as below - it is usual and convenient to distinguish between linguistic 'word' knowledge and conceptual 'world' knowledge. However, it seems somewhat difficult to maintain such a distinction consistently, because the language (i.e. the 'words') reflects concepts, objects and relations of the 'world'.

Some fundamental remarks related to the ETS

The ETS works with transfer based, stratificational translation principles. The grammar and the lexical rules are written in a formal language, the Eurotra formalism. A lexicon entry is a feature bundle (made up by the relevant features) containing the description of a given lexical item. Each feature is represented by an attribute/value-pair. A lexicon entry contains the linguistic knowledge about the lexical item: primarily morphosyntactic information types such as part of speech, inflection, syntactic frames, valency bound prepostions, etc. but no information about conceptual (non-linguistic or world) knowledge. The analysis and generation carried out by the ETS is based on observable, syntactic properties. It has been argued from different points of view in favour of integrating conceptual knowledge in NLP systems (e.g. in LILOG 1991). Furthermore, it should be metioned that within the Eurotra framework attempts have been made to introduce knowledge-based frames for terms (Selsoe Sørensen 1990).

The current work is based on the English-Danish translation module of the ETS including the appropriate lexicon components for general language. (The size is about 6000 entries for Danish). In the present version of the system, the analysis works with a combination of one English kernel dictionary component for general language and a satellite dictionary for the particular subject language. In synthesis a parallel set of Danish dictionaries is used. The interface between English and Danish is established by a transfer dictionary.

The main topic treated in the present project is the necessary extension and modification of the lexicon component for a selected domain-specific text corpus. The grammar component has been extended to cover syntactic phenomena which occur frequently in the chosen text type e.g. instructions given by imperative or modal constructions, nominalized structures (impersonal constructions), etc.

The dictionary component has been extended to cover (a chosen part of) the corpus, the new entries have been encoded with the standard Eurotra features required by the system, without integrating domain specific (world) knowledge.

The intention is to produce translation test output with an updated grammar and with an extended, but not modifyed dictionary covering the lexical items in the corpus. Consequently, in general erroneous translation and/or overgeneration are primarily caused by insufficient information about the lexical items in the dictionary.

Short outline of the work phases

The main point is to elaborate a method for description of lexical units for this domain which can be used as common basis for human and machine translation.

The main steps are:

1. Establishing the text corpus in machine-readable form: texts in both English and Danish were requested from a number of car companies and converting the printed material by scanning or keying in.
2. Generating automatically a concordance of the corpus (KWIC)
3. Analysis and lemmatization of the corpus and concordance as regards objects for lexical description: single word units and complex lexical units (multi-word units, compounds with special attention to proper boundary recognition). We are also looking at phenomena which may cause problems for human translators too.
4. Description of lexical units: elaboration of a draft version for coding schemes, with the main units being
   * linguistic information types needed for analysis/recognition of the source language text
   * subject information types needed for understanding and disambiguation (domain-specific language usage)
   * linguistic information types for production of target language text
   * subject information types to ensure the correct choice of domain-specific equivalents in translation.

Error sources and types

In analysis/understanding of the source language (English) the most common errors occur in the following areas:

* Recognition of multi-word units as regards their particular grammatical, syntactic and lexical structure and their non-compositional meaning
* Recognition of multi-word terms as such, which is very important for the correct translation from English into Danish compounds.
* Recognition of domain-specific usage of general words and expressions deviating from the domain-external usage, e.g. wrt. the valency bound preposition(s).

For correct text production in the target language (in our case in Danish) the translator makes use of information types besides the general production rules (grammar/morphosyntactics, lexicon), such as:

* Knowledge about the subject topic, which is often the key point for human translators and a lack in MT systems
* Equivalents for replacement of source language units, with special attention paid to lexical units, which have both general language and subject specific meaning(s), e.g. <u>washer</u> 'washing machine' vs. 'thin flat

ring' vs. 'equipment of a car washing windows', etc.

* Keeping track of the recognized multi-word terms and compounds to overcome the translation difficulties caused by erroneous segmentation or compositional analysis in the source language, e.g. the different ways of translating the word steering, contained as an initial element in a range of related compounds: 'steering arm, - box, - column, - column lever, - gear, - wheel' translates as 'spindel-, snekke-, rat-, and styre-'. The last compound translates into a single Danish (simplex) word: 'rat'.

Problems crucial both in human and in machine translation

When considering a number of translation problems common to human and machine translation, some particular error types occurring from lack of knowledge or information in the dictionary can be singled out.

Summing up: for successful translation of LSP (language for special purposes) texts, the knowledge of the source and the target language is essential, but not sufficient. Knowledge about the domain and the topic (world knowledge) is just as necessary for the understanding of the source text, which in turn is the essential prerequisite for producing the translation.

The linguistic knowledge in the areas of morphology, grammar etc. and general language knowledge is a basic requirement for all translation and is presupposed in the further considerations.

Investigations have been carried out for instance within the scope of the overall project on human translators' need for different information types. The investigations concerned both successful dictionary look-ups (Maidahl 1992) and erroneous translations (Møller 1992). The results show the overall tendency that for human translators the lack of knowledge and dictionary information about the domain is crucial. In our experience the lack of integrated world knowledge in the MT lexicon component causes similar problems.

Some fundamental observations on the text corpus led us to the assumption that the source text and accessible parallel texts in the target language contain a quantity of the needed conceptual knowledge. This means that, to some extent, knowledge about the subject field can be extracted from domain-specific texts. This is parallel to the fact that the domain specific (world) knowledge of human translators is often based on experience of using the particular language for special purposes (LSP).

In the analysis we use the following general strategy to extract knowledge about the domain and the sublanguage in question:

1. After lemmatization etc. (cf. item 3 above) a basic vocabulary for the corpus has been established.
2. The basic vocabulary has been investigated in the following way: we want to deal with the domain specific lexical items and for this purpose we have chosen a set of items.

2.1 For each chosen noun we extracted (all) corpus examples and established a list

of compounds with the search word as component. (The list then can be organized e.g. on the basis of the semantic role of the search word in the compounds.)
2.2. For each chosen noun and verb we extracted (all) corpus examples and established a list of collocations.
3. For each chosen item we have to collect information from the text, which can be relevant for the structuring of world knowledge, i.e. features which form 'part-of' and 'kind-of' relations between concepts, about the way objects are functioning etc. However, these derived pieces of information are not sufficient for describing concepts systematically, for instance in an ontological framework. Therefore, in case of lacking information, the pieces can be combined with descriptions extracted from other texts outside the corpus that belong to the same domain (e.g. Handbooks on car maintenance, etc).

Text based dictionary work
In the following, a few examples should illustrate some typical observations which provide useful aspects for text based dictionary work with which we are concerned

1. English compounds in LSP: autonomous word (spelled with blanks between) form a compound denoting one single concept or object. In the analysis the proper delimitation of the components which make up the compound (e.g. a multi-word term) can be quite difficult. The translation into Danish cannot be carried out compositionally at least in case of lexicalized compounds.
The whole text corpus contains e.g. 117 occurrences of the word wheel, approximately 85 times as a component of a compound (complex noun). The list below shows a selection of the most frequent types and examples:
Type 1.1: The search word wheel is the 1st component of a two-component compound:

-- brace, -- cap, -- change, -- changing, -- chock, -- disc,
-- nut, -- ornament, -- stud, -- trim...

Type 1.2: The search word wheel is the 1st component of a compound containing more than 2 components:

-- brace assembly, -- nut cover, -- nut wrench ...

Type 1.3: The search word wheel is not the 1st component of the compound:

front --s, driving front --s, rear --s, replacement --, road --, spare --, spare - - mounting bracket, steering -- ...
alloy --, aluminium --, steel -- (vs. plastic -- trim) ...
4-wheel drive (model) ...

Type 2: can be defined as a (p.t. not subdivided) list of frequent co-occurrences of the search word wheel (or a complex noun with the search word as one component) and a verb/verbal expression; here is only an illustrative choice of examples listed:

balance / change / chock / install / remove / ... a/the --.

The dictionaries for human users do not contain sufficient information about compounds. It means that the user probably can look up a complex lexical unit component by component, and will for each component find a list of possible

translations. The process of combining these translations to form an adequate expression in the target language will not be successful. Only a few domain specific multi-word terms and compounds are listed as lexicalized units in the examined bilingual dictionaries for the sublanguage (cf. Reference list). Nevertheless many of them cannot be understood compositionally, even if each component of such a complex lexical unit has an autonomous meaning. Therefore at least lexicalized compounds should be accessible in a sublanguage dictionary, but of course this will increase considerably the dictionary size.

2. Collocations specific to the sublanguage need a careful treatment too. In this case the main difficulty is the translation of the collocate, i.e. it has often a weak or deviating semantic content compared to general language use, for instance in the collocations 'apply the handbrake' vs. 'apply (...) pressure to the footbreak', i.e. the verb has to be translated in two different ways into Danish.

3. Ambiguous words and expressions within the sublanguage cause translational difficulties too (e.g. remove, release, replace etc.)

4. Ambiguous words with different translations in general language and in LSP. The last problem can be (and is often) solved in dictionaries.

We have at the present time extracted from the corpus a list of the most frequent nouns and verbs. Then we elaborated a comprehensive list of compounds, typical co-occurrences, collocations and meaning paraphrase for each interesting item using both the method of sequential search in the text and the elaborated KWIC concordance. The aim of this work is to set up a structured list of relevant units to be entered in the text based dicitonary.

Some requirements for the representation of information types

A short overview of available translation aids in this subject field shows the need for a comprehensive information source, as the lexical units of the examined bilingual dictionaries are very poor both in content and description of the lexical units. Handbooks and manuals offer the best domain specific information support on the topic. For machine translation, the subject domain information collected from suitable background material has to be formalized in the same way as the other (e.g. linguistic) information types.

The necessary content of coding schemes

The process of systematic analysis of the text corpus (especially the parallel texts in English and Danish) with regard to lexical units, terminology and general usage, as well as to the subject knowledge inherent in the source language texts, leads to the setting up different coding schemes. They are aimed at covering the information types described above, which are considered to be necessary or useful in human and machine translation. At the present time, we are working on the elaboration of the draft schemes, which include the following main information types (beginning at the source language side):

* Morphology (inflectional, derivational etc. properties)
* Syntax (part of speech, subcategorisational properties, valency bound

prepositions, etc.)
* Semantic properties (selectional restrictions, meaning description, etc.)
* Usage properties (collocations, idiomatic sense, etc.)
* Subject information (domain/subdomain, definition, explanation,etc.)
* Examples in English

The second part of the coding scheme covers the target language information types including translational equivalents in a similar way.

The standard dictionaries for technical sublanguages (cf. Dictionaries in the Reference list) provide in general only a simple listing of so called equivalents of source/target languages, without further information about linguistic and conceptual properties.

A brief discussion of some reults

We are presently working on refining the corpus analysis methods and the elaboration of coding schemes for different kinds of lexical units. The expected outcome of the sub-project will include detailed descriptions of the above discussed information types.

These descriptions provide the basis for elaborated dictionary entries, from which it is possible to automatically derive the appropriate database record for the EUROTRA translation system, formalized according the feature theory and extended with the necessary subject information types.

On the other hand, the same dictionary entry can be converted and extended for use by human translators. Such extensions are e.g. definitions, paraphrases and examples which are given in natural or metalanguage.

The objective is still to create a machine stored dictionary containing linguistic and subject field descriptions necessary for human and/or machine translation in a specific domain. Of course, the working method still has to be improved relative to further experience obtained from testing the model, as the ongoing project has arrived at the development and implementation phase.

References:

ARK 65: Oversættelse af fagsproglige tekster. [Translation of LSP texts]. Arnt Lykke Jacobsen (ed.) Copenhagen 1992. (Papers from the Project Meeting and Conference 1991)

Bech, A. & B. Maegaard: **Videnrepresentation i maskinoversættelse.** [Knowledge Representation in Machine Translation.] In: **ARK 65.**

Braasch, A.: **Valg af tekstsort - Korpus - Undersøgelsesaspekter.** [Selecting of Text Type - Corpus - Aspects of investigations.] In: **ARK 65.**

Copeland, C., J. Durand, S. Krauwer, B. Maegaard (eds.), **The Eurotra Linguistic Specifications**. Studies in MT and NLP, vol. 1. CEC, 1991.

Czap, H. & W. Nedobity (eds.), **TKE'90: Terminology and Knowledge Engineering**, vol. 1. & 2. Index Verlag Frankfurt/M., 1990.

Freibott, G. & u. Heid: **Terminological and lexical knowledge for computer-aided translation and technical writing.** In: **TKE'90**.

Gazdar, G. & Ch. Mellish: **Natural Language Processing in Prolog.** (p.306 ff.) Addison-Wesley, Wokingham 1989.

Gotlieb, C.C. & L. d'Haenens: **Machine Translation of Non-Literary Texts: Some Canadian Experiences** (p.21 ff)
In: Machine Translation Vol.6 - 1991 Kluwer Academic Publishers, Dordrecht.

KBMT-89 Project Report, Center for Machine Translation, Carnegie Mellon University, 1989.

Lang, Ewald: **The LILOG Ontology from a Linguistic Point of View.** In: Herzog, O. & C.-R. Rollinger (eds.): Text Understanding in LILOG: Integrating CL and AI. Springer Verlag, Berlin etc. 1991.

Maidahl, L.: **Brugen af fagordbøger. En empirisk undersøgelse.** [Using sublanguage dictionaries.] In: **ARK 65.**

Møller, B.: **Oversættelse af teknisk tekst. Anførte problemer og konstaterede fejl.** [Translation of technical texts. Problems and errors.]
In: **ARK 65.**

Selsoe Sørensen, H.: **The use of knowledge-based frames for terms in EUROTRA.** In: **TKE'90.**

Walton Evens, M. (ed.), **Relational models of the lexicon.** Cambridge University Press, 1988.

Dictionaries:

950 amerikanske-engelske automobil-fagudtryk oversat til dansk. [950 American and English terms.] General Motors International A/S (No furter information)

Teknisk ordbog. Engelsk-dansk. L&H Ordbøger. Copenhagen 1991.

Kjærulff-Nielsen, B.: **Engelsk-Dansk Ordbog.** Copenhagen 1985.

Skibsted, S.: **Teknisk engelsk-dansk ordbog.** Copenhagen 1971.

Skjerk, Ebbe: **Bilteknisk ordbog. Engelsk-Dansk.** [Dictionary for motor/car mechanics. English-Danish.] Teknisk Forlag 1991.

Warrern, A.: **Engelsk-dansk teknisk ordbog.** Copenhagen 1986.

Analyse des composes nominaux non lexicalises de l'allemand sur la base de la classe semantico-syntaxique de leurs constituants

DANIEL BRESSON

RESUME:

Les composés de l'allemand non lexicalisés, c'est-à-dire des unités polylexicales qui ne sont pas répertoriées dans les dictionnaires, représentent une partie importante des textes (1.) et posent des problèmes particuliers pour l'analyse automatique. En effet, ils sont formés par pure concaténation de deux unités lexicales, et c'est la connaissance des propriétés sémantico-lexicales des deux composants qui permet de déduire la nature de la relation entre A et B (2.). En 3. et 4., il est montré à l'exemple de la quantification comment l'étiquetage préalable des unités lexicales permet la détermination de la relation dans le composé.

1. LA NOTION DE COMPOSES NON LEXICALISES: L'EXEMPLE DE "Abfall"

Le dictionnaire Duden en un volume (Duden 1989) , qui compte 120000 items, mentionne sous l'entrée **"Abfall"** (déchet) trois composés avec **"Abfall"** comme premier composant: *-eimer* (poubelle), *-produkt* (*déchets, résidus*) et *-rohr* (sorte de *gouttière*). Le Duden en six volumes (Duden 1977), 500000 items, mentionne en plus de ces trois composés 10 termes supplémentaires: *-beseitigung* (*élimination des déchets*), *-erzeugnis* (production de), *-grube* (*fosse à ordures*), *-haufen* (*tas d'ordures*), *-korb*, *-kübel* et *-tonne* (*poubelle*), *-material* et *-stoff* (*résidus*), *-verwertung* (*utilisation des déchets*). Un dictionnaire spécialisé dans ce domaine (Seidel 1988) mentionne 41 composés possédant **"Abfall"** comme premier élément. Il s'agit essentiellement de termes spécialisés relevant des techniques de protection de l'environnement, et certains mots figurant dans les deux Duden n'y figurent d'ailleurs pas, comme *-eimer* et *-rohr*, car ils ne relèvent pas de cette technique.

D'autre part, une liste de noms établie à partir du corpus de l'IDS[1] fait apparaître 15 noms comprenant **"Abfall"** comme premier élément, mais seul dans cette liste *-verwertung* est mentionné dans Duden ou Seidel. Voici ces composés sur "Abfall" rencontrés dans des textes: *-entsorgung* (*élimination des déchets-* surtout radioactifs), *-entsorger* (*personne* ou *installation pour...*),*-vermeidung* (*éviter les*), *-definition* (*définition des*), *-konzept* (*notion de*), *-satzung* (*réglementation sur*), *-misere* (*catastrophe des*), *-beamter* (*fonctionnaire spécialisé dans*), *-experte* (*expert en*), *-management* (*gestion des*), *-händler* (*commerçant en*), *-rechtler spécialiste du droit de*).

On constate qu'apparaissent dans les textes des composés nominaux qui ne sont pour la plupart d'entre eux ni des composés répertoriés dans les dictionnaires généraux, ni des termes relevant du lexique de la spécialité, le corpus de l'IDS ne comprenant pas d'ouvrages techniques à proprement parler. Ces composés nominaux ont pour second composant des noms abstrait comme *Definition, Management, Konzept,* ou des noms d'agent tels *Experte, Beamter, Händler.* A première vue, ces composés ne relèvent pas du mode de formation de termes spécialisés tels que *Abfallbelebtschlamm* (boue active) ou *Abfalleinäscherung* (incinération des déchets), qui désignent des techniques particulières et figurent dans le dictionnaire spécialisé. Les premiers appartiennent au contraire au domaine général et sont formés selon des paradigmes très courants: A-*definition*, A-*konzept*, A-*experte*. Les composés des domaines spécialisés doivent être répertoriés dans des dictionnaires et traduits dans la langue étrangère en tant qu'unités

correspondant à un désigné identifiable. Il ne saurait être question d'essayer de les traiter comme des constructions libres et de leur appliquer les algorithmes que l'on utilise pour de telles constructions. Leur traitement relève du domaine de la terminologie plus que de l'analyse linguistique proprement dite. Mais on comprend inversement qu'il ne serait ni utile, ni facile d'essayer de faire des listes interminables d'unités lexicales ayant comme second composant des termes abstraits tels que *Definition*, *Konzept* ou *Experte*, qui peuvent être associés à un très grand nombre de mots en position A. Il me semble que cette constatation permet de bien situer et évaluer les problèmes posés par les composés lors du traitement automatique d'une langue comme l'allemand, qui offre à ses usagers la liberté de pouvoir concaténer en une seule unité graphique plusieurs unités lexicales sans former nécessairement pour autant une nouvelle unité lexicale. Cette liberté est largement utilisée par les locuteurs de l'allemand, et il faut donc s'attaquer au problème autrement, en tentant de d'analyser les relations possibles entre A et B[2]. Cette analyse concernera les unités nominales polylexicales qui n'ont pas le statut de terme de spécialité.

2. L'ANALYSE DE LA RELATION ENTRE LES CONSTITUANTS A ET B DANS LES COMPOSES NON LEXICALISES[3] : L'EXEMPLE DE "A-**Zeitschrift**" (**revue**).

Le dictionnaire général Duden (Duden 1989) mentionne les composés suivants formés sur "**Zeitschrift**" comme constituant B: *Fachzeitschrift* (*r. spécialisée*), *Frauenzeitschrift* (*r. féminine*) et *Literaturzeitschrift* (*r. littéraire*).

La liste de noms de l'IDS et la liste de mots inverse de ce même institut (Rückläufige Wortliste, 1986) comprennent en plus les mots suivants: *Gewerkschafts-* (*r.syndicale*), *Kinder-* (*r. pour enfants*), *Kunden-* (*r. pour les clients*), *Mode-* (*r. de mode*), *Rundfunk-* (*r.radiophonique*), *US-* (*r.américaine*), *Verbands-* (*r. de l'association*), *Verbraucher-* (*r. de/pour les consommateurs*), *Vertriebenen-* (*r. de rapatriés*), *Vieteljahres-* (*r. trimestrielle*), *Werk-* (*r. d'entreprise, r. "maison"*), *Wirtschafts-* (*r. économique*), *Wissenschafts-* (*r. scientifique*), *Wochen-* (*r. hebdomadaire*). On pourrait sans aucune difficulté allonger indéfiniment la liste en fabriquant des composés sur la base de "Zeitschrift" qui auraient l'air tout aussi authentiques que ceux-ci, qui sont attestés dans des textes.

Dans le volume IV de l'ouvrage "Deutsche Wortbildung" (Substantivkomposita, 1991), on trouve une typologie détaillée des relations dans des mots tels que ceux formés sur

"Zeitschrift". Les auteurs caractérisent chaque mot par son type et les rôles sémantiques des constituants A et B. Le type, ou la relation, est explicité au moyen d'une paraphrase. Le composé *Tierbuch* (*livre sur les animaux*), qui me semble correspondre du point de vue de sa structure à *Literaturzeitschrift* (revue littéraire) est analysé de la façon suivante (SK 132, 383-395):

Il appartient au groupe des "Bezugskomposita" de type "referentiell", définis par la relation "[B] betrifft/ bezieht sich auf [A]" (B se rapporte à/ concerne A). Le premier sous-ensemble, représenté par *Tierbuch*, correspond aux rôles sémantiques "thème / contenu - forme d'expression / représentation". Cette analyse me paraît tout à fait acceptable pour le cas étudié; et on pourrait aussi l'appliquer à l'analyse de *Literaturzeitschrift*: thème/contenu: la littérature; forme de l'expression (nous dirions plutôt: support): revue. Si l'on analyse aussi en suivant la méthode des auteurs de SK d'autres termes formés sur "Zeitschrift", comme *Wochenzeitschrift* ou *Werkzeitschrift*, on obtiendrait les résultats suivants:
Wochenzeitschrift (*r.hebdomadaire*): date - objet/moyen concerné (SK 499), avec une mention spéciale pour "date", qui devient [itératif] (SK 501)
Werkzeitschrift (*r.d'entreprise*): le type le plus proche semble être "agentif/auteur" (SK, 553), défini par les rôles sémantiques agent (auteur) - produit, bien que l'on puisse aussi analyser "*revue/journal d'entreprise*" comme "destiné à".

Un mot correspondant à un désigné unique, comme "revue" dans le cas qui nous occupe, va donc se trouver être investi de rôles sémantiques très variables selon les combinaisons dans lesquelles il va entrer. Le mode de fabrication des composés de l'allemand ne donne en général aucune indication sur la relation sémantique ou syntaxique existant entre les composants, contrairement à ce qui se passe en français pour les termes correspondants, où le choix de la préposition ou de la forme du terme modifieur permet de spécifier la relation: *revue musicale, pour les jeunes, de mode, sur les animaux* etc.. Or, ces rôles sémantiques possibles sont inscrits dans les gènes du désigné. Une revue fait partie de la classe d'objets "publications", caractérisée par les propriétés suivantes:

support/forme de l'objet (format, matière, qualité...)
fréquence de la parution (régulière,mensuelle..)
lieu/origine de la publication (française, régionale...)
éditeur/auteur: (personne, éditeur, institution, collectivité)
public visé (jeunes, consommateurs)
sujet/thème (scientifique, musicale, d'information...)

Chacune de ces propriétés peut être actualisée au moyen d' unités lexicales compatibles avec sa valeur:

format: *DIN-A4...*
qualité: *Luxus...*
fréquence: *Wochen, Vierteljahr*
lieu: *US*
éditeur / auteur: *Verband, Gewerkschaft*
public visé: *Verbraucher, Kinder*
sujet / thème / domaine: *Mode, Wissenschaft, Literatur*

Les propriétés sémantiques du premier constituant permettront de décider quelle est la relation qui sera d'actualisée. Cela ne lève pas toutes les ambiguités. Prenons l'exemple de "*Frauenzeitschrift*" (mot à mot "femmes-revue"). Le premier constituant *Frauen* peut théoriquement désigner l'éditeur (revue de femmes, faite par des femmes), le public visé (revue féminine), le sujet/thème (revue parlant des femmes). La troisième interprétation paraît peu probable, bien que des des revues aient pour thèmes "parents", "Mutter und Kind" (mère et enfant). Les ambiguïtés surgissent donc quand un terme peut permettre d'actualiser plusieurs relations, et seule une réflexion pragmatique et une certaine connaissance du monde permettent de donner à une interprétation une plus grande vraisemblance.

3. CLASSES SEMANTICO-SYNTAXIQUES[4]

La nature de la relation actualisée entre A et B dépend donc en premier lieu des propriétés combinatoires de ces mêmes éléments, qui sont elles-mêmes déterminées par l'appartenance de l'élément à une classe sémantico-syntaxique. C'est à ce classement que se sont attelés pour l'allemand des auteurs comme Dornseiff (Dornseiff, 1933, 1970), qui propose de classer conceptuellement les mots de l'allemand selon des classes d'objets ("begrifflich nach Sachgruppen ordnen"). Mais ce classement, s'il peut certainement être en partie repris, ne peut pas être utilisé tel quel, car il n'a qu'une base conceptuelle et repose essentiellement sur le champ sémantique auquel appartient un mot. On trouve dans le chapitre 13 (Dornseiff 363-382) consacré aux signes, à la communication et à la langue, un sous-chapitre 13.6 réservé à "informer, porter à la connaissance de" (*Bekanntmachen*), dans lequel on trouve les verbes, les noms, les adverbes, les expressions figées et imagées relevant de ce champ sans indication de la classe syntaxique. L'indication du domaine conceptuel est précieuse, mais elle n'est pas suffisante, il faut aussi savoir si le mot désigne un objet, un prédicat, un agent, un quantifieur etc...

Il faut donc aller plus loin dans dans la recherche et la description des classes. Une direction de recherche est fournie par G.Gross (Gross, 1990), qui établit des listes les plus exhaustives possibles de mots d'une même classe lexico-syntaxique (noms de fleurs, de boissons, d'actes juridiques etc.).

Mais il est nécessaire, parallèlement à cette recherche sur les classes lexicales terminales, de définir une topologie syntactico-sémantique: les notions de "substantif abstrait ou de substantif prédicatif" ont, en dehors de leur valeur sémantique, une signification pour la syntaxe de ces éléments. Un substantif prédicatif a une structure d'arguments donnée: Une *blessure* est infligée "par quelqu'un à quelqu'un, en telle ou telle occasion, sur telle ou telle partie du corps, au moyen de telle ou telle arme" etc..., et les composés nominaux allemands formés sur **"Verletzung"** (*blessure*) seront interprétés comme *blessure* **à** (*Kopf-, tête*), *blessure* **de** (*Krieg-, guerre*), *blessure* **par** (*Kugel-, balle*) selon que le terme A appartient à la classe des parties du corps, des événements ou activités, des armes ou objets (Bresson 1991, 186) et est susceptible, en fonction de cette appartenance, d'occuper telle ou telle position d'argument par rapport au prédicat **"Verletzung"**.

4. EXEMPLE DE CLASSE SEMANTICO-SYNTAXIQUE: LA QUANTIFICATION

Un certain nombre de composés nominaux allemands non lexicalisés sont caractérisés par le fait que l'élément B, au lieu d'être déterminé par l'élément A, est un nom de nombre, de quantité ou de mesure, qui opère comme quantificateur sur A; il peut s'agir de noms désignant des unités d'une certaine matière, tels que *Korn* (*grain*) ou *Molekül* (*molécule*), de noms désignant une certaine quantité de X ou un ensemble de X, tels que *Menge* (*foule, ensemble*) ou *Gruppe* (*groupe*). On peut distinguer un certains nombre d'opérations quantifiantes, susceptibles d'être exprimées par un ensemble d' unités lexicales qu'il importe de répertorier et d'affecter dans le dictionnaire d'un symbole servant à signaler leur valeur d'opérateur quantifieur; il existe souvent un ou plusieurs mots classifieurs représentatifs de tout le groupe. Opérateurs identifiés:

qun (quantifieur unité)= une unité de A; mots classifieurs: ***Einheit, Element***; ex: *Produktionseinheit* (*unité de production*), *Bauelement* (*unité de construction*); autres mots appartenant à cette classe: *Atom, Ion, Keim (germe), Molekül, Organismus, Partikel, Teilchen* (*particule*), *Zelle* (*cellule*) etc..

qstr (quantifieur structurant)= une structure de A; ces mots servent à désigner la structure habituelle d'apparition d'une matière donnée; mot classifieur? mots appartenant à cette classe: *Schicht* (*couche*), *Vorkommen* (gisement), *Staude* (*pied de*), *Stück* (*morceau de*, pour le sucre par exemple) etc..

qocc (quantifieur occurrence de)= un cas de A; mot classifieur: *Fall* (*cas de*); ex: *Ausnahmefall* (*exception*), *Krankheitsfälle* (*des cas de maladie*); autres termes: *Phänomen;*

qpar (quantifieur partie de)= une partie non structurée de A; mot classifieur: *Teil* (*partie de*); ex: *Bevölkerungsteil* (*partie de la population*); autres termes: *Spur* (*trace*), *Anteil* (*part*), *Abschnitt* (*passage*), *Bruchteil* (*parcelle*), *Splitter* (*bris*), *Stelle* (*passage*)

qsstr (quantifieur sous-structure)= une partie structurée de A; mots classifieurs: *Glied* (*membre*), *Teil* (*pièce, partie de*); ex: *Motorteil* (*pièce du moteur*) autres termes: *Absatz* (*paragraphe*), *Kern* (*noyau*) etc.

qgr (quantifieur un certain nombre de): un certain nombre d'unités A dénombrables; mot classifieur: *Gruppe;* ex: *Kindergruppe* (*groupe d'enfants*); autres termes (il y en a plusieurs centaines): *Schar* (*grand nombre*), *Menge* (*foule*) etc..

qq (quantifieur une certaine quantité de): une quantité non mesurée de A non comptable; mot classifieur: *Masse;* ex: *Lavamassen* (*masses de lave*); autres termes: *Menge, Haufen* (*tas*)

qn (quantifieur nombre de): spécifie une valeur numérique de A quantifiable; mot classifieur: *Zahl;* ex: *Übersiedlerzahl* (*nombre de rapatriés*); autres termes: *Werte* (*valeurs*), *Ziffern* (*chiffres*), *Dosis* (*dose*), *Statistik* etc.

qcol (quantifieur totalité ou collectif): désigne la totalité de A; mots classifieurs: *Gesamtheit* (*totalité*), *Bestand* (*l'ensemble*); ex: *Fichtenbestand* (*l'ensemble des épicéas*); autres termes: *Allgemeinheit, Ganze* (ensemble), *Ladung* (*chargement*), *Vorrat* (*réserve*), *Komplex* (*ensemble*) etc.

qm (quantifieur mesure): indique les mesures de A; il se subdivise en plusieurs opérations selon le type d'indication de mesure correspondant à l'objet désigné par A; chaque dimension/mesure est exprimée au moyen d'unités de mesure appropriées.

qdim (quantifieur dimensions): désigne les dimensions de A; mots classifieurs: *Ausmaße, Dimension (dimensions), Umfang (volume)*;ex: *Zimmerdimensionen (dimensions de la pièce)*; autres termes: *Höhe (hauteur), Breite (largeur), Länge (longueur), Fläche (surface)* etc.

qh (quantifieur niveau): désigne le niveau, le taux de A; mots classifieurs: *Niveau, Pegel (niveau)*; ex: *Ozonpegel (niveau d'ozone)*; souvent exprimé en valeur relative et non absolue comme **qdim**; autres termes: *Satz (taux), Rate (pourcentage), Höhe (hauteur)* etc.

qsom (quantifieur somme): désigne une certaine somme de A, souvent monétaire; mot classifieur: *Summe (somme)*; ex: *Investitionssumme*; autres mots classifieurs: *Betrag (montant), Zuschuß (subvention), Gebühr (taxe), Abgabe (prélèvement)* etc.

qd (quantifieur temporel durée): s'applique à des A quantifiables par leur durée; mot classifieur: *Dauer (durée)*; ex: *Geltungsdauer (durée de validité)*; autres termes: *Zeit (temps, durée), Periode* etc.

qfr (quantifieur temporel fréquence, vitesse): indique la fréquence ou le rhythme ou la vitesse de A; mot classifieur: *Frequenz, Rhythmus, Geschwingigkeit* (vitesse); ex: *Arbeitsrhythmus (rythme de travail)*; autres mots: *Tempo, Häufigkeit* etc.

qdis (quantifieur spatial distance): concerne des A de valeur spatiale; mot classifieur: *Entfernung*; ex: *Grenzentfernung (distance de la frontière)*; autres termes: *Distanz, Abstand (espace, distance)* etc.

qs (quantifieur scalaire, intensité): indication du degré, de l'intensité de A; mots classifieurs: *Grad (degré)*, Intensität; ex: *Verstrahlungsgrad (degré d'irradiation)*; autres termes: *Skala (échelle), Stufe (degré)* etc.

qdif (quantifieur différence, comparaison): mesure la différence entre des valeurs de A; mot classifieur: *Unterschied (différence)*; ex: *Höhenunterschiede (différences d'altitude)*; autres termes: *Gleichheit (égalité), Ungleichheit (inégalité)* etc.

qv (quantifieur variations): spécifie les variations de A; mot classifieur: *Variation, Schwankung*; ex: *Klimavariationen*; autres termes: *Verlust (perte), Abnahme (diminution), Erhöhung (augmentation)* etc.

5. CONCLUSION ET PERSPECTIVES

L'exemple de la quantification, qui représente une relation assez bien identifiable entre deux composants A et B dans le cas de composés nominaux de l'allemand, avait pour objectif d'illustrer comment l'établissement de classes sémantico-syntaxiques dans les unités lexicales répertoriées d'une langue peut être utilisé pour déterminer la relation sémantique susceptible d'être réalisée lors de la mise en connexion de deux unités de la langue. Les composés non lexicalisés de l'allemand sont un cas extrême, puisque dans ces constructions, aucune marque morphologique ne permet de déduire le type de relation syntaxique existant entre A et B. Il s'agit d'une pure concaténation, et la mise en rapport des deux unités par la concaténation est le seul indice apparent utilisable pour déchiffrer la valeur de la construction. Mais cet indice a une valeur si générale qu'il est inutilisable. Il signifie simplement : "il y a une relation entre A et B". D'où la nécessité de chercher, dans les propriétés de A et de B, les éléments de signification qui peuvent justifier que ces deux unités lexicales soient mis en relation. Cela suppose un très important travail de description des unités qui permette de prévoir leur comportement combinatoire. C'est à ce prix que l'on peut espérer certains progrès dans le traitement automatique de langues agglutinantes comme l'allemand.

NOTES

1. La liste à laquelle il est fait allusion ici est une liste alphabétique électronique de près de 200000 noms établie à partir du corpus de l'Institut für Deutsche Sprache de Mannheim et qui a été très généreusement mise à ma disposition par cet institut. Pour tous renseignements concernant ce corpus, cf REFER (Brückner 1986).

2. Les composés allemands sont fondamentalement de structure binaire, quel que soit le nombre des unités lexicales dont ils sont constitués et toute analyse, à l'excetion de quelques rares structures ternaires (type bleu-blanc-rouge), peut donc être ramenée à l'aétude de la relation entre un constituant A et un constituant B.

3. Je laisserai de côté dans cette communication le cas des composés formés sur un composant B de type "relationnel / rectionnel" ou prédicatif. Dans ce cas, ce sont les propriétés combinatoires du terme B qui ouvrent un certain nombe de structures ou de relations possibles, et les caractéristiques sémantique de A permettent de décider de la relation actualisée par une association A-B donnée (Bresson 1991)

4. Pour une représentation d'ensemble de cette question cf Bresson, 1992

BIBLIOGRAPHIE

BRESSON, D.(1991) "Zur Analyse nominaler relationaler Komposita im Deutschen im Hinblick auf die maschinelle Sprachverarbeitung", *Cahiers d'Etudes Germaniques 21*, 179-188, Aix-en-Provence

BRESSON, D.(1992) "La relation syntaxique et sémantique dans les composés nominaux de l'allemand", *Systèmes interactifs, Mélanges en l'honneur de Jean David*, 67-79, Metz

BRÜCKNER,T. (1986) *Refer, Benutzerhandbuch*, Institut für deutsche Sprache, Mannheim

DUDEN (1978) *Das große Wörterbuch der deutschen Sprache in 6 Bänden*, Bibliographisches Institut Mannheim

DUDEN (1989) *Deutsches Universalwörterbuch A-Z, Bibliographisches* Institut Mannheim

GROSS, G. (1990) "Les classes d'objets", *Actes du colloque sur les industries de la langue*, Montréal

Rückläufige Wortliste zum heutigen Deutsch, 2.Auflage (1986) bearbeitet von T.BRÜCKNER und C.SAUTER, Institut für Deutsche Sprache Mannheim

SEIDEL,E. (1988) *Wörterbuch Umweltschutztechnik*, Verlag Harri Deutsch, Thun und Frankfurt(Main)

SK (1991), *Deutsche Wortbildung, Vierter Hauptteil: Substantivkomposit*, de Gruyter,Berlin, New York

Dictionary Completeness and Corpus Analysis

DAVID CLEMENCAU

ABSTRACT

We present here an experiment of corpus analysis by electronic means. This experiment was made on a French corpus of 1,300,000 occurrences which correspond to 58,000 graphically different words.

In the first phase of this analysis, we parsed our corpus with several linguistic filters including a large dictionary of inflected words, a dictionary of compound words and local grammars. The second phase was the analysis of the remaining - that is, unrecognized by the filters of the first phase - words by a two-level morphological analyzer which turned to be very efficient in recognizing and analyzing derivatives.

This experiment led to the enrichment of both dictionaries of simple and compound words and of the list of names of the LADL. It also led to a significant enhancement of our two-level analyzer which should become a reliable tool when filled with all the French inflectional classes.

*Laboratoire d'Automatique Documentaire et Linguistique: 2, place Jussieu; 75251 Paris cedex 05; France.
**Institut Gaspard Monge: Université Marne la Vallée; 2, allée Jean Renoir; 93160 Noisy le Grand; France.

I. INTRODUCTION, MOTIVATIONS

Dictionary completeness is a fundamental issue when analyzing large corpora. When a word encountered in a text cannot be found in an electronic lexicon, the analysis of the sentence is highly compromised. It has been shown by Courtois B., Laporte E. 1991 that, when a simple word appearing in a text is not found in the dictionary of simple inflected words DELAF[1], it is often a derived word. Crude morphological analysis may sometimes provide grammatical information for such unknown words. But we preferred a different approach, aiming at a fine description of the derivational paradigm of the entries of the LADL lexicon-grammar of French verbs[2] (see Clemenceau D. 1992). For practical reasons, we are engaged in two different procedures:

- the complete syntactic description of 12,000 verbs has complex implications for the morphological processes. Taking fully advantage of this information will take some time[3]. Moreover, the description of the derivational behavior of nouns and adjectives must also be done;

- we construct a morphological analyzer based on the two-level model (Koskenniemi K. 1983, 1984) which avoids the failure of the dictionary look-up process for derived words but which provides only partial information.

This analyzer is embedded in a tool for corpus analysis that we present in this paper. We experimented this tool on a corpus of 1,300,000 occurrences of French words. This experiment had two main goals:

- dictionary enrichment, by:

  a) adding new words in the dictionary of simple words DELAS;
  b) adding names to the list of names of the LADL;
  c) recording rules of derivation to be included in our description of the derivational behavior of French verbs;

- the design of a tool that tags the words of a text, avoiding, as much as possible, "lexical holes" that jeopardize the full analysis of texts.

II. METHOD

When analyzing texts, i.e. strings of ASCII characters, one usually splits up the ASCII table into two parts: alphabetic characters and non alphabetic characters, i.e. potential separators

---

[1]DELAF stands for Electronic Dictionary of Inflected Forms of the LADL. It contains about 700,000 entries and is automatically generated from DELAS, the dictionary of simple forms, which contains about 80,000 entries in their canonical form. (Courtois B. 1984)

[2]The lexicon-grammar of verbs can be seen as a matrix of 12,000 rows (the verb entries) and 300 columns (the syntactic properties). For each entry a "+" or a "-" in each column indicates wether it satisfies the associated syntactic property. Moreover, a sub-categorization of entries into tables, according to their main construction, has been designed (Gross M. 1968, 1975 ; Boons J.P., Guillet A., Leclere C. 1976a, 1976b).

[3]At the present time, we have tested our method of describing derivational paradigms on the entries of four tables of the lexicon-grammar only.

for words, paragraphs... A word is then defined as a string of alphabetic characters without any internal separator, bounded by two separators. However, since we wanted our tool to treat derived words such as *sous-traiter* as units, we had to consider that hyphens between two alphabetic characters are not separators. According to this definition of words, we isolated 1.300.000 occurrences in our corpus which correspond to 58,000 graphically different words.

This list of 58,000 words has been analyzed by several linguistic filters as shown in figure 1. Each filter of our tool extracts the words that it recognizes (dotted arrows) and sends the unrecognized words to the next filter (solid arrows).

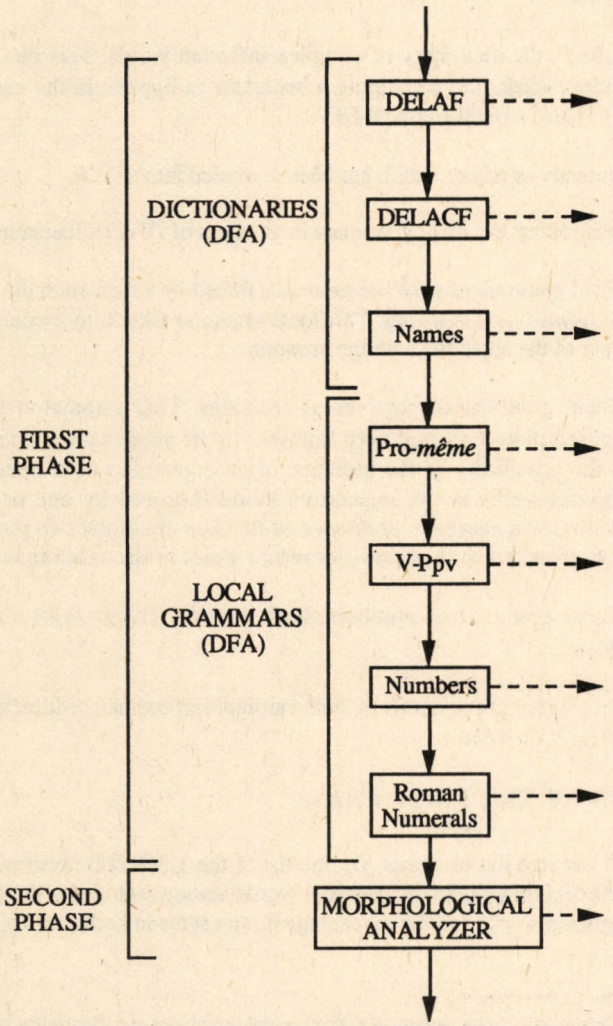


Figure 1: the structure of the corpus analysis tool

We split the analysis process into two phases in order to clearly distinguish the heuristic part of our tool, i.e. the morphological analyzer. The first phase of the analysis is composed of:

- DELAF, the dictionary of simple inflected words. This dictionary has been compiled into a multi-terminal deterministic finite automaton[4] in order to achieve a low memory cost (less than 2 Mb for a dictionary of 700,000 entries with their morphological code, including the look-up software) and a very fast look-up procedure[5].
  When a word contains capitalized letters, which often loose their diacritic signs in French, the look-up software looks for it in lower case, reconstructing all possibilities for the corresponding words with diacritics (Roche E. 1992). A word like *ELEVE*, for example, will be recognized as the capitalized form of both *élève* and *élevé*.

- DELACF, the dictionary of complex inflected words. This dictionary of 130,000 complex words, i.e. containing a separator (a hyphen in the case we are treating here), is also a multi-terminal DFA.

- a dictionary of names which has been compiled into a DFA.

The four following filters are local grammars in the form of DFA (Silberstein M. 1989):

- the local grammar of personal pronouns linked by a hyphen to the adjective *même* as in *toi-même* or *elles-mêmes*. This local grammar takes into account the agreement in number of the adjective with the pronoun.

- the local grammar of post-verbal pronouns. This automaton recognizes strings composed of an inflected verb followed by its subject pronoun as in *voulez-vous*, with the possibility of the insertion of an euphonic *t* as in *acceptera-t-il*. It also recognizes verbs in the imperative mood followed by one or two complement pronouns as in *mange-le* or *donnez-la-lui*. The use of the two pronouns *en* and *y* in such expressions as in *penses-y* or *mènes-y-moi* is also taken into account.

- the local grammar of numbers, both cardinal (*vingt-sept*) and ordinal (*vingt-septième*).

- the local grammar of numbers, both cardinal and ordinal, written in Roman numerals (*XXVII*, *XXVIIème*).

III. RESULTS OF THE FIRST PHASE

We parsed our two list of words, i.e. the list of the 1,300,000 occurrences in our corpus and the list of the 58,000 graphically different words among them, with the DELAF filter. The number of recognized words and the percentage it represents in both cases is given in figure 2.

---

[4]This type of deterministic finite automata (DFA) which combines the flexibility and the quickness of lexicographic trees with the possibility of being minimized has been introduced by Revuz D. 1991.

[5]The look-up procedure of the 1,300,000 occurrences of our corpus takes about 13 minutes, dictionary loading time included, on a PS/2 i386DX 25MHz with the OS/2 1.3 system. The average rate of the look-up procedure is thus 100,000 words per minute.

	OCCURRENCES	WORDS
TOTAL	1,328,126	57,697
DELAF WORDS	1,275,613	41,760
% DELAF	96%	72%

Figure 2: results of the DELAF filter

At the sight of these figures, two observations arise:

- our corpus is mainly composed of occurrences of entries of DELAF (96%) ;
- the 15,000 words of our corpus that are not in DELAF have an average rate of appearance of 3.3 whereas the rate of appearance of the 42,000 words found in DELAF is 30.5.

The DELAF filter gave also the number of words of DELAS used in our corpus. For each word it recognizes, the DELAF filter returns the words in DELAS that could be its canonical form. For instance, the word *président* will be recognized and will cause the program to return the following string:

président,président.N32:ms,présider.V3:P3p:S3p

which means that *président* can be either the singular masculine form of the noun *président* or the third person of plural of the verb *présider* in the present indicative or subjunctive tense. By collecting the returned value of the DELAF filter for the 42,000 words of our corpus that it recognized, we were able to build the sub-DELAS of our corpus which contains about 20,000 entries. Thus, a quarter of DELAS was used in our corpus.

At the end of phase one, we got a list of 14,367 words that were not recognized by any filter. This list can be split into a list of 2,504 complex words, i.e. containing a hyphen, and a list of 11,863 simple words. These lists are mainly made of names, foreign words (mainly English), typing errors or spelling mistakes, compound words containing a hyphen and derivatives.

In phase two, which is described in the next part of this paper, we experimented our morphological analyzer on these remaining lists.

IV. THE MORPHOLOGICAL ANALYZER

A. Two-level morphology

In his 1983 thesis, Kimmo Koskenniemi proposed a declarative system of constraints for describing morphological phenomena (Koskenniemi K. 1983). These constraints are called two-level rules. The main difference between two-level rules and rewrite rules, introduced in the 1960s by (Chomsky N., Halle M. 1968), is their static nature. For instance, the following rewrite-rule:

*a -> b / c_*

means that the underlying symbol *a* is rewritten as the surface symbol *b* in the environment following *c*. There are two important consequences of the dynamic nature of this type of rule. First, after underlying *a* is written as surface *b*, *a* is no longer available for any other rules. Second, rewrite rules can only apply in one direction, from underlying representation to surface representation. They cannot be reversed to go from surface to underlying representation. On the contrary, the corresponding two-level rule:

*a:b <= c:c_*

express the relationship between the underlying (lexical) *a* and the surface *b* as a static correspondence. Rather than rewriting *a* as *b*, this two-level rule states that a lexical *a* corresponds to a surface *b* in the environment following *c:c*, which is also a lexical-to-surface correspondence. This rule does not change *a* into *b*, so *a* is still available to other rules[6].

There are two consequences of the static nature of two-level rules:

- in a two-level system, rules are not ordered. Each rule can be compiled into a finite-state transducer and at each step of the recognition (resp. generation) process, all the rules are applied in parallel. An equivalence between an underlying symbol and a surface symbol will be allowed if it is recognized by all the rules, i.e. by all the transducers of the rule set.

- two-level rules can be applied for both recognition and generation of surface forms.

B. A two-level analyzer for productive derivation rules

The prototype of morphological analyzer we built was originally designed for the recognition of derivatives. This prototype is based on the PC-Kimmo implementation of Koskenniemi's system (Antworth E.L. 1990). This system is composed of two components: the lexicon and the set of rules. The lexicon can be seen as a tree whose branches represent morphemes, each path from the root of the tree to a terminal leaf corresponding to a valid lexical form. When a surface form is analyzed, for each surface symbol (from left to right) all possible correspondences between surface and lexical symbols that can apply are analyzed by the two-level rules. If a given correspondence satisfies all the two-level rules, the resulting lexical form is compared to the lexicon. If this lexical string correspond to a valid path in the lexicon, the correspondence is allowed and the system goes to the next surface symbol to the right. The last symbol of the generated lexical form must leads to a final leaf in the lexicon for the lexical form being a valid analysis.

Since the PC-Kimmo implementation was not designed to handle such a lexicon as DELAS, we used is lexicon file in a special way: our lexicon contains affixes but no stem. Rather than filling the lexicon with a small part of DELAS, we chose to fill the lexicon with the skeleton of a DELAS word, i.e. one or more alphabetic characters[7]. The structure of our

---

[6]This opposition between dynamic and static rules can be linked with the opposition in computer science between procedural and declarative programming.

[7]For such an implementation of the two-level model, see *Parsing with a minimum lexicon* in (Antworth E.L.

lexicon is given in figure 3, where the labels mean:

- *PREFIX*: a list of about 100 French prefixes. Among them, we recorded 30 morphemes which are not derivational prefixes: prefixes derived from country names as *afro-* (*Afrique*) and prefixes derived from adjectives as *électro* (*électrique*). Thus, our analyzer recognizes some compound words too.

- *DELAF ENTRY*: a lexical morpheme corresponding to this branch is considered as valid only if it is found in DELAF.

- *DELAS ADJECTIVE, VERB, NOUN*: a lexical morpheme corresponding to these branches is considered as valid only if it is found in DELAS with the same category.

- *ADJECTIVES, VERBS, NOUNS INFLEXION*: inflectional suffixes of adjectives and nouns (feminine and plural) and verbs (conjugation). We took into account only a few classes among the 80 inflectional classes of adjectives and nouns and the 100 classes of verbs that are recorded in DELAS. We will see that this lack is the main cause of over-recognition in our system. However, the integration of all these classes would mean that DELAS, DELAF and the mechanism of inflection that links both of them in both way (i.e. generation of inflected forms from the canonical form and retrieval of the canonical(s) form(s) corresponding to an inflected form) would be handled by the two-level system. This does not seem feasible in the PC-Kimmo framework. However, recent works in the two-level domain seem to propose a integration of the lexicon and the set of rules into a unique transducer that could lead to such a system (Karttunen L., Kaplan R., Zaenen A. 1992).

- *ADVERB FORMATION*: formation of adverbs from adjectives, using the suffix *-ment* (*actif -> activement*).

- *N -> V*: suffixes that transform a noun into a verb. We recorded the following suffixes: *-iser* (*catégorie -> catégoriser*), *-ifier* (*classe -> classifier*), *-er* (*groupe -> grouper*).

- *Adj -> V*: suffixes that transform an adjective into a verb. Beside the three suffixes *-iser*, *-ifier* and *-er*, we recorded *-ir* (*grand -> grandir*).

- *V -> Adj*: present participle (*aimer -> aimant*), past participle (*aimer -> aimé*) and the suffix *-able* (*aimer -> aimable*).

- *V -> N*: nominalization suffixes, i.e. *-ation* (*animer -> animation*), *-age* (*caler -> calage, blanchir -> blanchissage*) and *-ement* (*lancer -> lancement, blanchir -> blanchissement*).

- *Adj -> N*: the suffix *-ité* (*actif -> activité, activable -> activabilité*).

As for the rules set, we took into account most of the morphological phenomena recorded

---

1990 237-238).

in (Guilbert L. 1971), which led to a set of 40 rules.

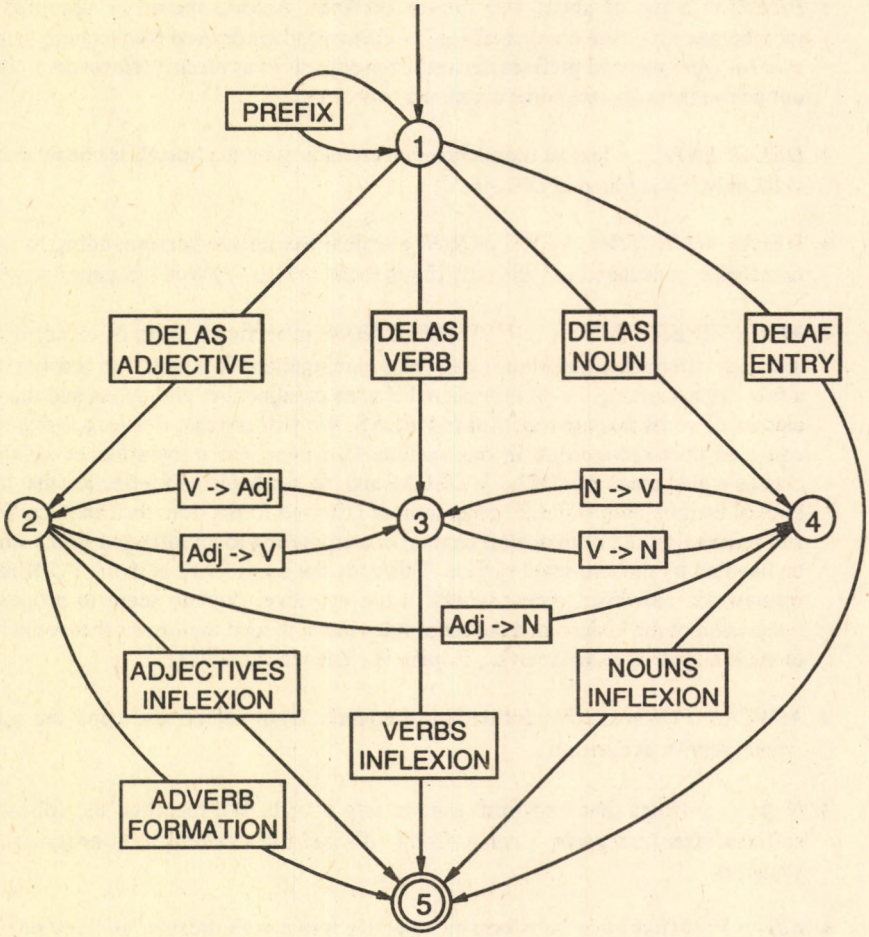


Figure 3: the structure of the lexicon

C. Results of the second phase

We applied two different versions of our prototype to the list of words that was not recognized during the first phase. First, we applied a version whose lexicon contained only words of DELAF with one or more prefixes[8]. We were able to recognize 900 of the 2,568 words containing a hyphen with this version. As for the list of simple words, 307 words were recognized among 11,863. In these successful analyses, we found very few over-recognition problems, i.e. non-valid analyses.

[8]This version corresponds to a simplified version of the lexicon of figure 3 that contains only the two branches *PREFIX* and *DELAF ENTRY*.

The second experiment was made with the full lexicon of figure 3 that enables suffixes analysis. We recorded very good results for derivatives. Some examples of analyses are given in figure 4.

```
débureautisation:
     dé_+bureau+ise+5tion:   [D(dé-) + bureau.N3 + N->V(iser) + Vn(ation)]
contextualisent:
     contextuel+ise+ent:     [contextuel.A40 + Adj->V(iser) + P3p]
     contextuel+ise+ent:     [contextuel.A40 + Adj->V(iser) + S3p]
indissociabilité:
     in_+dissociabilité:     [I(in-) + dissociabilité,dissociabilité.N21:fs]
     in_+dissociable+ité:    [I(in-) + dissociable.A31 + Adj->N(ité)]
     in_+dissocie+able+ité:  [I(in-) + dissocier.V3 + Vable + Adj->N(ité)]
     in_+dissociabilité:     [I(in-) + dissociabilité.N21]
     indissociable+ité:      [indissociable.A31 + Adj->N(ité)]
```

Figure 4: some examples of analyses

However, many spelling mistakes were recognized as valid words because of our poor description of inflections. For example, in the analysis of figure 5, the verb *aller* was treated as a regular verb whereas it is highly irregular. Both analyses do not correspond to the surface form. The first one, second person of singular in the present subjunctive tense, corresponds to the surface form *ailles* and the second one, second person of singular in the present indicative tense, to *vas*.

```
alles:
     alle+es:                [aller.V16 + S2s]
     alle+s:                 [aller.V16 + P2s]
```

Figure 5: over-recognition

V. CONCLUSION, FUTURE ACTIONS

The first phase of this experiment led to a substantial enrichment of both DELAS and DELAC. Moreover, it produced a large list of proper names. It gave also statistical information about the frequency of use of the DELAS words by building the sub-DELAS of our corpus, i.e. the words of DELAS that appeared in our corpus in any inflected form.

In the near future, we are planning to complete our syntactic description of the derivational behavior of French verbs, which should lighten the work of our morphological analyzer, and thus limit the heuristic part of our corpus analysis tool. We are also engaged in improving this tool, in order to provide a reliable basis for the syntactic analysis of large corpora.

As for our morphological analyzer, we are investigating the recent developments in two-level domain which seem to propose implementations that could realize a full two-level DELAF, i.e. a two level system including both DELAS and DELAF with generation and recognition of inflected words capacity. We think that such a system, working together with a set of two-level rules handling derivational processes should lead to a very efficient morphological analyzer.

V. REFERENCES

Antworth E.L. 1990: *PC-KIMMO: a two-level processor for morphological analysis.* Occasional Publications in Academic Computing N°16, Summer Institute of Linguistics, Dallas, Texas.

Boons J.P., Guillet A., Leclere C. 1976a: *La structure des phrases simples en français (constructions non complétives) I-Les verbes intransitifs.* Genève,Droz.

Boons J.P., Guillet A., Leclere C. 1976b: *II-Classes de constructions transitives.* E.R.A. N°247 du C.N.R.S.

Chomsky N., Halle M. 1968: *The sound pattern of English.* Harper and Row. New York.

Clemenceau D. 1992: *Problèmes de couverture lexicale* in Langue Française, Productivité et créativité lexicale. Larousse, Paris. (to appear)

Courtois B. 1984: *Le dictionnaire des formes simples du français.* Rapport de recherche du LADL, Université Paris VII.

Courtois B., Laporte E. 1991: *Une expérience de dépouillement de textes : les mots non reconnus.* Autogen, Genelex project.

Gross M. 1968: *Grammaire transformationnelle du français, syntaxe du verbe.* Paris, Larousse.

Gross M. 1975: *Méthodes en syntaxe, régime des constructions complétives.* Paris, Hermann.

Guilbert L. 1971:*Fondements lexicologiques du dictionnaire. De la formation des unités lexicales,* in GLLF, t. I.

Karttunen L., Kaplan R., Zaenen A. 1992: *Two-level morphology with composition.* COLING'92, 141-148.

Koskenniemi K. 1983: *Two-level morphology: a general computational model for word-form recognition and production.* Publication N°11. Helsinki: University of Helsinki, Department of General Linguistics.

Koskenniemi K. 1984: *A general computational model for word-form recognition and production.* COLING'84, 178-181.

Revuz D. 1991: *Dictionnaires et lexiques. Méthodes et algorithmes.* Thèse de doctorat, Université Paris VII.

Roche E. 1992: *Building compacted dictionaries experiment.* Technical report, Institut Blaise Pascal. (to appear)

Silberstein M. 1989: Dictionnaires électroniques et reconnaissance lexicale automatique. Thèse de doctorat, Université Paris VII.

COMET: un Système informatique de génération de métaphoresen langue française

CORNU GÉRALD — HÜE JEAN-FRANÇOIS — SIMON YVES — WALLE JEAN-MICHEL

Abstract :

COMET is a computer system which was developed at the University of Nantes to generate metaphors in French using a lexicon and a set of fixed model sentences. Given a word as user input, it uses a rule base to work out other words which can be associated to it in order to build metaphorical sentences. The present version of COMET has a lexicon of about 150 words and generates 5 types of metaphorical sentences. It uses the object-oriented SMALLTALK language.

The problem of generating metaphors is twofold. On one hand we have to define possible metaphorical relationships between lexical domains, and on the other hand we have to arrange the lexical content within these domains so that we can map them with one another and define metaphorical relationships between words having an analogical position within their respective domains.

Our approach involves an unconventional reworked version of Lakoff and Johnson's theory of metaphor, and a semantics inspired by structuralism, case grammar and narrative theory.

* *
*

1 - Présentation de COMET

COMET est un prototype mis au point à l'Université de Nantes dont la finalité est de tester des hypothèses linguistiques sur le mécanisme de la métaphore. En laissant de côté l'analyse pour ne se concentrer que sur la génération d'énoncés métaphoriques, COMET permet de réduire au minimum l'appareillage morpho-syntaxique et sémantique que l'on trouve habituellement dans les systèmes de traitement automatique des langues naturelles pour consacrer l'essentiel des efforts à la définition des relations qui doivent exister entre deux termes afin de produire un effet métaphorique.

Ces termes apparaissent comme des variables pouvant s'insérer à l'intérieur de modèles de

phrases figées. Nous noterons X un terme de départ entré par l'utilisateur et qui sera obligatoirement un nom. Selon les modèles de phrases choisis, le système aura à rechercher dans le lexique :

- un terme X', également un nom, mais appartenant à un domaine lexical différent de celui de X et pouvant entrer en correspondance métaphorique avec X.

- un terme V' : il s'agira d'un verbe appartenant à un domaine lexical différent de celui de X et qui, associé à lui dans un énoncé produit un effet métaphorique (V' se "substitue" à un verbe V appartenant au domaine lexical de X et qui peut être virtuel, c'est-à-dire qu'il n'a pas d'expression lexicale).

- un terme Y : il s'agit d'un nom appartenant au même domaine lexical que X et dont le rôle est de compléter certains modèles d'énoncés comprenant déjà X et X', ou X et V'.

- un terme Y' : il s'agira d'un adjectif appartenant à un domaine lexical différent de celui de X et dont l'association avec X produit un effet métaphorique (comme V', Y' se substitue à un adjectif Y appartenant au domaine lexical de X et qui peut être virtuel).

Dans la notation utilisée ici, X, V et Y appartiennent à un même domaine lexical D qui sera dit domaine de départ. X', V' et Y' appartiennent à un domaine lexical D' dit domaine d'arrivée.

Les modèles de phrases utilisés dans COMET sont actuellement au nombre de 5 :

(1) "Le X est le X' du Y."

Exemple : "Le dessert est la conclusion du repas."

(2) "Le X est un X' pour le Y."

Exemple : "La lecture est un aliment pour l'esprit."

(3) "Le X V' le Y."

Exemple : "La lecture rassasie l'esprit."

(4) "C'est un X très Y'."

Exemple : "C'est un livre très indigeste."

(5) "Ce X est un vrai X'."

Exemple : "Ce livre est un vrai régal."

D'autres modèles de phrases pourront bien sûr être implémentés dans COMET, l'objectif étant d'insérer les variables dans un contexte linguistique naturel permettant au linguiste de juger la validité des relations entre variables découlant des règles programmées dans le système plus facilement que s'il avait affaire à de simples listings.

Le principe de COMET est celui d'une démarche interactive entre le système et l'usager-linguiste, permettant à celui-ci de tester des règles de déduction de termes X', V' et Y' à partir d'un terme X de départ, et de les affiner progressivement au vu des outputs générés par le système.

2 - Les bases théoriques

Le problème de base de la génération de métaphores est celui de la déduction d'un (ou plusieurs) terme X' à partir d'un terme X (ou V' à partir de V, ou Y' à partir de Y) traduisant la possibilité d'établir un lien métaphorique entre les deux termes. Ce lien métaphorique permettra soit l'association des deux termes dans un énoncé métaphorique (métaphore *in praesentia*), soit la substitution de l'un par l'autre de ces deux termes (métaphore *in absentia*).

Comme nous l'avons vu plus haut, chaque terme du lexique est partiellement défini par son appartenance à un domaine lexical D. Le problème à résoudre est alors double :

- d'une part, il s'agit de déterminer à partir d'un domaine lexical de départ D quels sont les domaines lexicaux D' dont les termes peuvent entrer en correspondance métaphorique avec des termes de D.

- d'autre part, il s'agit de déterminer à partir d'un terme X appartenant à D quels sont les termes X' appartenant à D' susceptibles de s'associer ou de se substituer métaphoriquement à X.

2.1 - Les relations métaphoriques entre domaines lexicaux

Un domaine lexical D pourra être défini en première approche comme un ensemble structuré de morphèmes se rapportant à un même domaine d'activité. Ainsi, dans les exemples utilisés plus haut, "dessert", "repas", "aliment", "rassasier", "indigeste" et "régal" appartiennent à un domaine lexical que nous nommerons "alimentation". "Livre", "lecture" et "conclusion" appartiennent au domaine de la "lecture", que nous pourrons enrichir de termes tels que "lecteur", "avant-propos", "chapitre", etc.

Ces domaines lexicaux correspondent, dans une terminologie plus usuelle en lexicologie, à ce que Lakoff et Johnson (1980) appellent des "concepts". Avec eux, nous considérons que le mécanisme de la métaphore ne s'applique pas à des morphèmes pris isolément, comme le laissait penser une tradition fortement ancrée en Europe qui faisait de la métaphore, comme de la métonymie et de la synecdoque, des "tropes", c'est-à-dire de simples figures de mots, mais à des ensembles de morphèmes susceptibles d'être mis en relation. La métaphore permet de parler d'un "concept" dans les termes d'un autre concept. Certains exemples classiques ("Argument is War", ou "Time is Money") analysés par ces auteurs permettent de rendre compte d'un grand nombre de métaphores courantes autant en français qu'en anglais.

Dans les exemples que nous avons donnés, la relation entre les "concepts" (domaines lexicaux) de l'"alimentation" et de la "lecture" justifie la production d'énoncés métaphoriques où des termes appartenant au lexique de l'alimentation sont associés ou substitués à des termes appartenant à celui de la lecture. La relation entre ces deux domaines possède une grande productivité métaphorique dont il serait facile de multiplier les exemples :

"Max est un lecteur vorace."

"Max est affamé de lectures."

"Il se gave de lecture."

"Certains chapitres de ce livre sont savoureux." Etc...

Face à la théorie de Lakoff et Johnson, notre divergence essentielle consistera à remettre en cause le caractère conventionnaliste de leur approche. En effet, ces auteurs ne retiennent qu'une liste finie de relations entre concepts, et cette liste est établie empiriquement. Il nous semble

possible d'aller plus loin dans la compréhension des mécanismes métaphoriques en recherchant un principe rationnel justifiant l'établissement d'un lien métaphorique entre deux domaines lexicaux. Pour que ce lien existe, il faut selon nous qu'il y ait une analogie globale entre ces domaines. Nous constaterons ainsi que les domaines de l'alimentation et de la lecture ont en commun l'existence d'un transfert effectué de l'extérieur vers l'intérieur d'un être humain. Dans le cas de l'alimentation, ce transfert a un caractère matériel, et dans celui de la lecture, il a un caractère spirituel. D'autres domaines lexicaux, comme celui de l'"apprentissage", partagent ce même caractère de transfert de l'extérieur vers l'intérieur, et la possibilité de générer des métaphores entre ces domaines semble aller dans le sens de notre hypothèse :

"Il a soif d'apprendre."

"Il ingurgite une quantité étonnante de connaissances."

A l'inverse, certains domaines que nous nommerons provisoirement "création", "accouchement", "invention", "germination" semblent relever d'une catégorie de transfert de l'intérieur vers l'extérieur, et offrent également des possibilités de métaphores :

"Il a une imagination fertile."

"Il a accouché d'un chef-d'oeuvre."

"L'idée a germé dans son esprit."

En revanche, aucune métaphore n'est possible entre des domaines relevant de ces deux catégories opposées, comme l'"alimentation" et la "germination" ou la "création".

Nous pourrons procéder de la même façon pour les domaines de la "guerre", de la "discussion", du "sport", de la "politique", etc., qui relèvent tous d'une catégorie plus générale du "conflit". Cependant, la richesse et la variété des lexiques se rapportant à la notion de conflit nous conduisent à envisager une hiérarchie de domaines lexicaux allant du général vers le spécifique. Le conflit se subdivise en conflit humain ou non-humain, puis parmi les conflits humains en conflits collectifs ou individuels, puis par exemple parmi les conflits individuels en conflits de type violent ("lutte") et conflits de type pacifique ("discussion", "sport individuel", "jeu de compétition").

Nous observons également que le vocabulaire du conflit se répartit entre tous les niveaux de l'arborescence ainsi construite et pas seulement au niveau inférieur des domaines lexicaux correspondant à des actions spécifiques. Certains morphèmes sont applicables aussi bien à des conflits individuels qu'à des conflits collectifs (par exemple "attaquer", "attaque", "combat", "victoire"...). D'autres ont un caractère encore plus général et transcendent l'opposition violent/pacifique ("gagner", "perdre", "adversaire"...). D'autres, enfin, ne s'appliquent qu'à un domaine limité (la "guerre" : "belligérants", "état-major", "bataille", "ultimatum"... ; la "discussion" : "argument", "réplique", "objection"...).

L'hypothèse à laquelle nous aboutissons est celle d'une hiérarchie en arbre (avec la possibilité de treillis par endroits) de domaines lexicaux "pleins", c'est-à-dire qu'ils sont tous susceptibles, quel que soit leur niveau dans l'arborescence, de contenir des morphèmes.

L'établissement de liens métaphoriques entre ces domaines lexicaux dépendra d'un parcours dans l'arbre. Tout domaine, quel que soit son niveau, pourra être pris comme domaine de départ, et l'on considérera qu'il n'y a pas métaphore lorsque le domaine d'arrivée est un domaine fils ou ancêtre du domaine de départ : il s'agit alors de lexiques plus ou moins spécifiques se rapportant à un même type d'activité. En revanche, il y aura métaphore possible avec tous les domaines ayant un père ou un ancêtre commun avec le domaine de départ. Cette ascendance commune traduit en termes plus rigoureux l'impression d'"analogie" pouvant exister

entre un domaine d'arrivée et un domaine de départ.

Une telle approche, encore hypothétique car elle demande à être affinée sur plusieurs points, s'écarte de celle de Lakoff et Johnson dans la mesure où elle propose un critère formel et non plus empirique de définition des correspondances métaphoriques entre "concepts", et où elle offre une analyse plus fine de la répartition des morphèmes en différents domaines lexicaux. Elle s'en écarte également dans la mesure où nous ne retenons pas le principe établi par ces auteurs d'une univocité de la relation entre domaines de départ et domaines d'arrivée. Ce principe souffre en effet de nombreuses exceptions. Par exemple, si le lexique de la "guerre" fournit des métaphores au domaine de la "discussion", l'inverse vaut aussi dans certains cas. Il est tout aussi acceptable d'utiliser les verbes "répliquer" ou "répondre" dans un énoncé se référant à la guerre, que les verbes "contre-attaquer" et "riposter" dans un énoncé concernant la "discussion". Cependant, il est bien évident que le principe de totale réciprocité des correspondances métaphoriques entre domaines appliqué dans la version actuelle de COMET conduit à générer de nombreuses phrases inacceptables, et que seule une analyse fine portant sur un grand nombre d'outputs permettra d'établir des règles rendant mieux compte de l'intuition linguistique des locuteurs d'une langue.

En s'écartant de la théorie de Lakoff et Johnson qui définit les "concepts" empiriquement et isolément les uns des autres et en privilégiant une structuration hiérarchique des domaines lexicaux, notre hypothèse se rapproche des théories sémantiques componentielles à orientation structuraliste. En effet, le raisonnement que nous avons suivi peut être reformulé en se plaçant non plus du point de vue des classes d'objets (les domaines lexicaux), mais du point de vue des objets eux-mêmes (les morphèmes). Chaque morphème étant partiellement défini par son appartenance à un domaine, cela peut se traduire par la possession d'un sème (trait sémantique) commun à l'ensemble des morphèmes d'un domaine lexical. Ce sème distingue les morphèmes appartenant à un domaine donné des morphèmes appartenant à un domaine frère. En revanche, tous les morphèmes appartenant à des domaines frères ont en commun une même ascendance que l'on pourra marquer par la possession de sèmes correspondant aux domaines père, grand-père, etc. Chaque morphème possède ainsi une série de sèmes plus ou moins longue selon que l'on descend dans l'arbre des domaines lexicaux, et qui permettent de le situer par rapport aux autres morphèmes.

Dans cette conception, les sèmes représentent en quelque sorte l'"étiquette" des domaines lexicaux, et la hiérarchie des domaines lexicaux équivaut à une hiérarchie de sèmes au sens que Greimas (1966) donne à cette notion, c'est-à-dire à un réseau d'identités et de différences où chaque sème se définit non pas "en soi", mais de façon différentielle par sa place dans un système d'oppositions pertinentes similaire à celui qui distingue les phonèmes en tant qu'entités formelles en phonologie. Cette conception structuraliste de la sémantique a d'ailleurs été appliquée à l'analyse des métaphores, puisque le Groupe Mu (1970) définit la relation entre les deux termes d'une métaphore comme celle de deux ensembles de sèmes possédant une intersection (un ou plusieurs sèmes communs) et un reste dans chacun des deux ensembles. Cette définition concorde avec celle que nous avons donnée plus haut dans les termes d'un parcours à travers l'arbre des domaines lexicaux, puisque les morphèmes susceptibles de produire un effet métaphorique possèdent en commun des sèmes désignant un ou plusieurs ancêtres communs à leurs domaines respectifs, et se différencient par d'autres sèmes qui marquent la différence de leurs domaines d'appartenance.

En tirant tout le parti de cette conception différentielle du sens, il nous est possible d'aller plus loin dans la définition des domaines lexicaux en adoptant un critère formel et non plus intuitif pour fixer leur extension, et c'est la métaphore elle-même qui nous fournira ce critère.

Reprenons l'exemple du domaine de l'"alimentation", et demandons-nous si le lexique concernant la cuisine (le fait de cuire les aliments) doit ou non lui être incorporé. L'approche référentielle désignant les contours d'un domaine lexical en référence à un champ d'activité dans un monde réel ou possible ne nous offre pas de solution. Il y a en effet un lien évident entre le fait de cuisiner et le fait de manger, mais cela ne suffit pas à indiquer si les lexiques associés à ces deux activités doivent entrer dans un même domaine lexical. L'approche structurale nous

conduit à nous interroger sur les oppositions pertinentes qui différencient un domaine par rapport à d'autres domaines lexicaux. Comme nous l'avons vu, le lexique concernant l'action de manger permet des métaphores avec les domaines de la lecture et de l'apprentissage. Cela signifie qu'ils ont un sème (ou ancêtre) commun que nous avons désigné sous le terme de "transfert de l'extérieur vers l'intérieur", et des sèmes différents qui marquent leur appartenance aux domaines différents de l'"alimentation", de la "lecture" et de l'""apprentissage". En revanche, le lexique associé à la cuisine ne permet aucune métaphore avec les domaines de la lecture et de l'apprentissage, mais il se prête à des métaphores avec des lexiques impliquant une idée de transformation d'un objet ("réflexion intellectuelle"...). Nous en déduirons que le lexique de la cuisine relève d'un domaine lexical distinct, que ses morphèmes ne possèdent pas d'intersection sémique avec ceux appartenant aux domaines pré-cités, et que le domaine de la "cuisine" s'inscrit donc dans un arbre séparé par rapport à celui des transferts de l'extérieur vers l'intérieur.

Nous avons utilisé ici la métaphore comme un critère permettant de définir les domaines lexicaux de façon différentielle les uns par rapport aux autres. S'il n'y a pas de métaphore possible entre deux lexiques, cela signifie soit qu'ils appartiennent à un même lignage (il y a inclusion d'un ensemble de sèmes dans un autre et l'un des deux ensembles n'a pas de reste), soit qu'ils relèvent d'arbres séparés (il n'y a pas d'intersection entre les deux ensembles de sèmes). S'il y a métaphore possible, cela signifie que les lexiques appartiennent à un même arbre, et que leurs compositions sémiques manifestent à la fois des identités (ancêtres communs) et des différences qui permettent de les situer l'un par rapport à l'autre.

Le procédé que nous venons d'employer est analogue à celui qui permet aux phonologues de définir les phonèmes d'une langue en tant qu'entités abstraites et selon un critère formel. L'épreuve de commutation consiste en effet à rechercher des paires de morphèmes où la seule variation d'un élément phonétique entraîne une variation du sens. Les différences phonétiques susceptibles de faire varier le sens permettent de définir des oppositions pertinentes entre phonèmes, cependant que d'autres différences restant sans influence sur le sens des mots ne représentent que des réalisations différentes d'un même phonème (par exemple l'opposition entre le "a" d'avant et le "a" d'arrière en français moderne). L'équivalent rhétorique de l'épreuve de commutation, consistant à rechercher s'il existe des métaphores possibles entre deux lexiques, nous semble permettre de fixer la position et l'extension des domaines lexicaux selon un critère rationnel solide. Il nous permet de renoncer à l'intuition empirique et de redéfinir les domaines lexicaux comme des entité abstraites à valeur explicative comme le sont les phonèmes en phonologie.

Cette hypothèse nous semble offrir des débouchés assez riches pour la lexicologie et la sémantique, et c'est l'un des intérêts de COMET que de permettre de la tester. Il est intéressant de rappeler que Jakobson et les autres fondateurs de la phonologie moderne regrettaient de devoir faire appel à un critère extrinsèque, en l'occurence sémantique, pour définir les phonèmes. Ceux qui après Hjelmslev fondèrent la sémantique structurale sur le modèle de la phonologie en "homologuant" les plans du contenu et de l'expression souhaitèrent également s'en tenir à des critères internes à la sémantique pour définir les unités minimales de sens que sont les sèmes. Il est peut-être logique que la sémantique fasse appel à la rhétorique pour fixer ses concepts de base là où la phonologie fait appel à la sémantique.

Précisons pour finir que les sèmes dont il a été question jusqu'ici sont exclusivement des sèmes génériques (correspondant aux semantic markers de Fodor et Katz) et non des sèmes spécifiques (distinguishers) dont le rôle consiste à différencier les morphèmes à l'intérieur d'une classe. D'autres figures de rhétorique comme la métonymie (CORNU, 1991, p. 480) et peut-être aussi la synecdoque, la litote et l'hyperbole, pourraient fournir à la définition des sèmes spécifiques des critères analogues à celui qu'offre la métaphore pour les sèmes génériques.

2.2 - La structuration des domaines lexicaux

L'autre partie du problème posé par la génération d'énoncés métaphoriques consiste, une fois que l'on a trouvé un domaine D' pouvant entrer en correspondance métaphorique avec un domaine D, à déterminer quel morphème X' (ou V' ou Y') appartenant à D' peut être associé ou substitué métaphoriquement à un morphème X (ou V ou Y) appartenant à D.

Pour cela, il nous faut définir un principe de structuration des domaines lexicaux permettant d'effectuer un "mapping" (CARBONELL, 1982) entre ces domaines, c'est-à-dire de faire correspondre terme à terme les éléments de ces domaines. Nous n'avons pas suivi la démarche structuraliste classique qui aurait consisté à poursuivre à l'intérieur des domaines lexicaux la hiérarchie sémique qui régit les relations entre les domaines. Ce principe se heurte vite à des problèmes insurmontables face à la richesse du matériel lexical d'un domaine qui contient, rappelons-le, des noms, des verbes, des adjectifs et des adverbes.

L'autre option consistait à utiliser comme principe de structuration la grammaire des cas de Fillmore. Cette option est beaucoup plus réaliste, mais elle laisse néanmoins en suspens un certain nombre de questions. S'il est vrai que les domaines mentionnés jusqu'à présent paraissent pouvoir s'ordonner selon un schématisme casuel faisant intervenir des agents, instruments, lieux, manières, etc., et si l'homologation de ces positions paraît en mesure de justifier les transferts métaphoriques, la grammaire des cas trouve ses limites lorsqu'il s'agit de rendre compte de relations qui ne sont pas seulement casuelles mais aussi séquentielles. Dans un domaine comme celui de la "guerre", il sera difficile de rassembler dans une même classe des morphèmes aussi différents que "attaquer", "contre-attaquer", "résister", "vaincre", "occuper", d'autant plus que les agents, instruments ou objets liés à ces actions ne sont pas nécessairement les mêmes.

Ces remarques nous conduisent à proposer une organisation narrative et non strictement casuelle des domaines lexicaux. Nous nous rapprochons ainsi des réseaux de type "script" ou "frame", qui décrivent aussi une succession organisée d'actions. Cependant, la nécessité de pouvoir homologuer les structures des différents domaines pour y discerner des positions similaires de termes devenant ainsi métaphorisables rend ces schématismes insuffisants. Ils sont en effet trop dépendants des connaissances référentielles spécifiques à chaque domaine, et ne répondent pas à l'exigence d'une forme indépendante des contenus qui s'y investissent et préservant l'unité du domaine lexical quelle que soit la diversité des actions et des acteurs qui lui sont associés.

Notre choix se portera sur une théorie de la narrativité plus abstraite, tout en restant proche à la fois d'une théorie différentielle du sens et, comme l'a montré Umberto Eco (1988, p. 118), de la grammaire des cas de Fillmore : il s'agit de la sémiotique des actants du récit d'A. J. Greimas (1966, 1970, p. 249 sq.). Selon cette théorie, qui est une formalisation de la théorie plus ancienne de V. Propp (1965), tout récit peut être conçu comme le passage d'un énoncé d'état (disjonction du sujet et de l'objet) à un autre énoncé d'état (conjonction du sujet et de l'objet) à travers une série d'étapes organisées autour d'une action principale, celle qui permet le passage d'un état à un autre. A partir de cette matrice de base qui définit le sujet comme sujet de "désir" et l'objet comme objet de "valeur" à l'intérieur d'un projet dynamique et irréversible, les autres actants viennent prendre leur place selon le rôle qu'ils occupent par rapport au "projet narratif". C'est ainsi que la modalité du pouvoir définira l'adjuvant comme celui qui facilite l'action du sujet, et l'opposant comme celui qui y fait obstacle. La modalité du savoir définit le destinateur comme celui qui suscite l'action du sujet prenant alors le rôle du destinataire. L'anti-destinateur suscite au contraire l'action d'un anti-sujet dont le désir vient contredire celui du sujet.

Les actants ainsi définis pourront être complétés par des cas tels que l'instrument, le lieu ou le signe, et l'on constate aisément que certains des actants mentionnés ci-dessus correspondent à des cas traditionnels comme l'agent, l'objet, le contragent. La structure des domaines lexicaux a simplement été enrichie d'un grand nombre de positions nouvelles, et l'on peut maintenir l'hypothèse que la relation métaphorique présuppose une position identique de deux termes dans leurs domaines respectifs. Le domaine de l'alimentation comportera par exemple une position

initiale (faim, soif), et une série d'étapes impliquant un certain nombre d'actants humains ou non-humains, qui conduisent le sujet vers son "rassasiement". De même, le domaine de la guerre partira d'une hostilité des protagonistes, et s'orientera, à travers la déclaration de guerre, l'attaque, la contre-attaque, l'assaut final, etc. (qui sont autant de termes susceptibles de transferts métaphoriques), selon une visée de la victoire qui est l'objet de désir du sujet. Comme on l'observera facilement, certains termes du lexique demeurent stables sur toute la durée, ou sur une partie importante du récit, cependant que d'autres ne valent qu'à un moment déterminé et dans une situation déterminée (actant-cas) par rapport à l'action.

Concrètement, la position d'un morphème dans un domaine pourra être déterminée par un système de 4 coordonnées $b_j c_k d_l e_m$. La première coordonnée indique à quel moment du récit et pour quelle durée le morphème trouve à s'appliquer (ex. "faim" : état initial ; "dessert" : fin d'action principale). La seconde coordonnée indique le statut actanciel assumé par le morphème (ex. "ennemi" : anti-sujet ; "renfort" : adjuvant). La troisième coordonnée indique si l'on a affaire à l'actant de base ou à un type d'action, d'attribut ou de lieu qui lui est spécifique (ex. "vindicatif" : attribut spécialisé d'un anti-sujet "adversaire" ou "ennemi" ; "tirer" : action spécialisée associée à un instrument "arme à feu"). La quatrième coordonnée indique la catégorie grammaticale et rend ainsi compte des nominalisations ou adverbialisations (ex. "assaillir"-"assaut" ; "agressif"-"agressivement").

Chaque position ainsi définie par ses quatre coordonnées peut correspondre à un ou plusieurs morphèmes. Globalement, la relation métaphorique peut être établie lorsqu'il y a identité de position entre deux morphèmes dans leurs domaines respectifs, c'est-à-dire lorsque la valeur des quatre coordonnées est égale, mais un certain nombre d'aménagements permettent de prendre en compte la possibilité pour X ou X' d'occuper plusieurs positions telles que sujet et anti-sujet (ex. "combattant"). Par ailleurs, beaucoup de positions demeurent inoccupées. Dans la version actuelle de COMET, chaque domaine comporte environ 1500 positions possibles dont la majorité sont bien entendu vides. Cela correspond au caractère lacunaire des langues naturelles où beaucoup de notions "pensables" ne sont pas lexicalisées, et cela nous amène à souligner l'un des aspects les plus intéressants de la productivité métaphorique : l'utilisation d'une métaphore correspond très souvent à la nécessité de combler un vide lexical en allant rechercher un équivalent dans un autre domaine. L'un des intérêts de COMET est de pouvoir coder des "places vides" et étudier le fonctionnement de métaphores *in absentia* dans des modèles de phrases tels que "Le X V' le Y", ou "C'est un X très Y'", où le verbe V et l'adjectif Y ne sont pas nécessairement lexicalisés.

Nous ne nous étendrons pas ici sur les règles permettant de déduire à partir d'un morphème X des morphèmes Y ou V appartenant au même domaine lexical. Ces règles ont pour but de faciliter l'utilisation de COMET en limitant l'input à un seul terme, et elles consistent simplement à coder des relations permettant de générer des énoncés sémantiquement cohérents.

2.3 - Remarques

Le système COMET permet d'analyser les mécanismes de la métaphore en testant et en affinant progressivement des hypothèses linguistiques sur la production métaphorique. Les difficultés rencontrées actuellement concernent le codage des morphèmes entrés dans le lexique. Chaque domaine lexical analysé pose de nouveaux problèmes pour assigner à certains morphèmes une place dans le système de coordonnées existantes. Ce codage est destiné à évoluer progressivement tout en conservant le principe d'une organisation narrative des lexiques qui en assure la cohérence. Par ailleurs, l'ajout de nouveaux domaines conduira à un degré de complexité beaucoup plus important dans la structure des arbres et dans la mise en relation de ces arbres.

Cependant, le problème essentiel posé par ce système réside dans sa trop grande puissance. COMET fait fonctionner le processus de la métaphore comme une combinatoire globale dont aucune langue naturelle n'utilise le potentiel dans sa totalité. Cette combinatoire n'est pas sans

bases réelles. On la retrouve en psychanalyse dans l'analyse des phénomènes de l'inconscient qui, tels le rêve, la névrose ou les actes manqués, font jouer la totalité des correspondances possibles entre éléments signifiants. Cependant, les langues naturelles font chacune un usage limité de leur potentiel métaphorique, et bien des métaphores théoriquement possibles seront jugées inacceptables par les locuteurs de ces langues. A terme, COMET devra inclure des mécanismes permettant de réduire sa puissance pour se rapprocher de la productivité métaphorique effective du français, tout en reconnaissant que la génération automatique ne pourra jamais épouser totalement cette productivité qui ne dépend pas seulement de mécanismes structurels, mais aussi d'éléments contingents et parfois éphémères, liés à l'histoire de la langue et à l'histoire des sociétés dans lesquelles elle est utilisée.

3 - Le logiciel de génération de métaphores

COMET permet de bâtir l'environnement de travail: lexique, règles, domaines lexicaux, types de phrases et de travailler à partir de cet environnement en choisissant le mode de recherche et en générant les métaphores.

3.1 - Bâtir l'environnement de travail

Le logiciel permet à l'utilisateur linguiste de travailler dans les meilleures conditions possibles aussi bien pour enregistrer ses données que pour vérifier ses hypothèses.
Pour cela il fournit toutes les possibilités de mise à jour (ajout, modification, supression) aussi bien au niveau lexical, qu'en ce qui concerne les règles gérées dynamiquement en PROLOG, les domaines lexicaux ou encore les types de phrases générées.

3.2 - Choisir le mode de recherche

Il est sélectionné par le logiciel à partir du domaine de départ D (par exemple la "discussion") et des mots choisis dans ce domaine (par exemple "débat", "argument") ainsi que du type de phrase choisi par l'utilisateur linguiste (par exemple "ce X est un vrai X'.").

3.3 - Bâtir les métaphores

Le logiciel peut travailler soit à partir d'un domaine d'arrivée D' choisi par l'utilisateur linguiste (COMET vérifie alors que ce domaine D' est un domaine d'arrivée possible), soit en cherchant tous les domaines d'arrivée D' possibles.
Dans les deux cas le logiciel va chercher tous les mots de substitution possibles par rapport aux mots du domaine de départ et pour finir générer les phrases conformément au type désiré.
Pour reprendre les exemples donnés au paragraphe précédent:
- s'il est proposé comme domaine d'arrivée D' la "guerre",le logiciel vérifie que la position de ce domaine d'arrivée peut bien fournir des métaphores par rapport au domaine de départ; il cherche alors tous les mots de substitution possibles par rapport aux mots du domaine de départ.
Pour les mots donnés en exemple, il fournirait, entre autres:
"Ce débat est une vraie bataille."
"Cet argument est une vraie déclaration de guerre."
-sinon COMET recherche tous les domaines d'arrivée D' possibles, il trouve, bien sûr, la "guerre", mais aussi, par exemple, d'autres "conflits".
COMET donnerait donc, après traitement, par exemple, en plus des phrases générées précédemment:
"Ce débat est un vrai pugilat."
"Cet argument est un vrai coup bas."

Bibliographie :

CARBONELL J. (1982), "Metaphor : an inescapable phenomenon in natural language understanding", in Lehnert & Ringle : *Strategies for natural language processing*, Hilscale.

CORNU G. (1991), "Quelques propositions pour le traitement des métaphores en TALN, in *Actes du Congrès Informatique et Langues Naturelles (ILN'91)*, Nantes, LIANA.

ECO U. (1988), *Sémiotique et philosophie du langage*, PUF.

GREIMAS A.J. (1966), *Sémantique structurale*, Larousse.

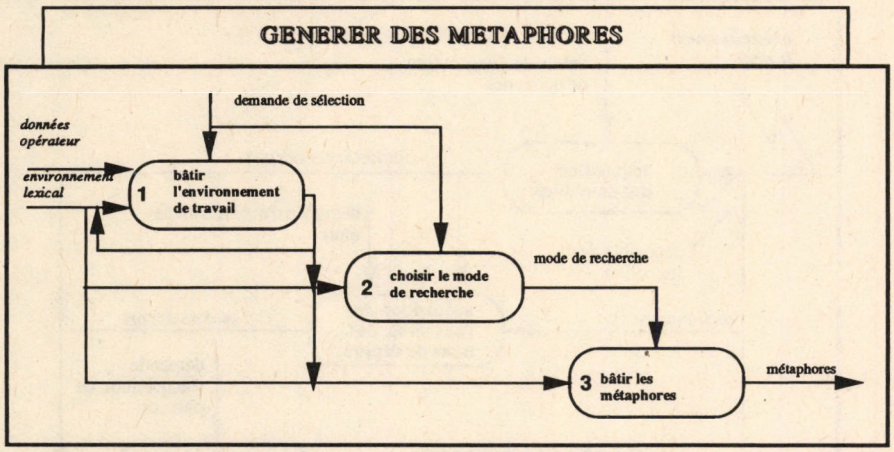
GREIMAS A.J. (1970), *Du sens 1*, Seuil.

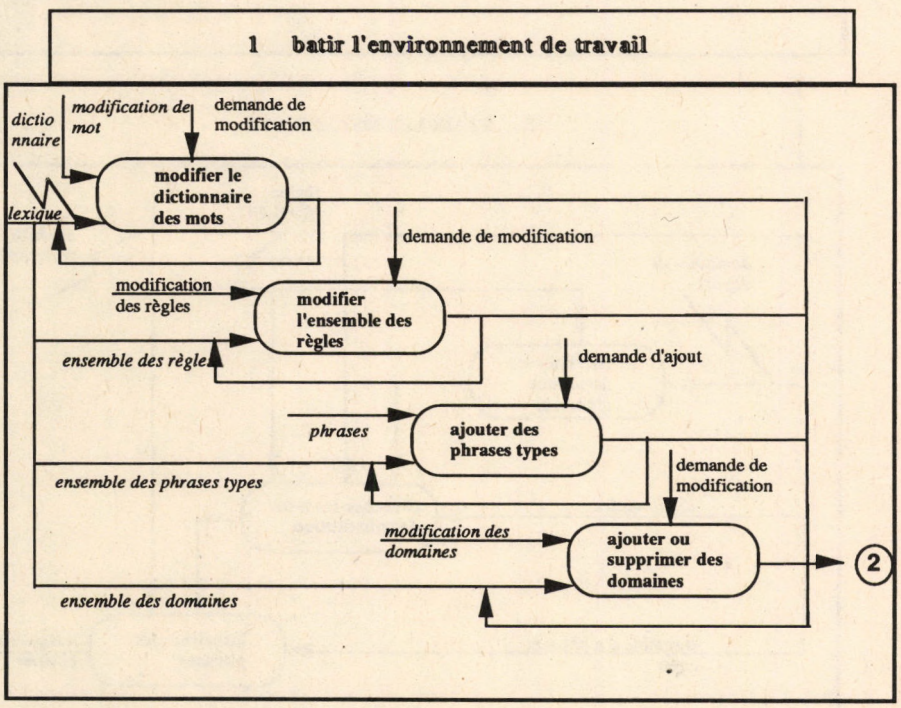
GROUPE MU (1970), *Rhétorique générale*, Larousse.

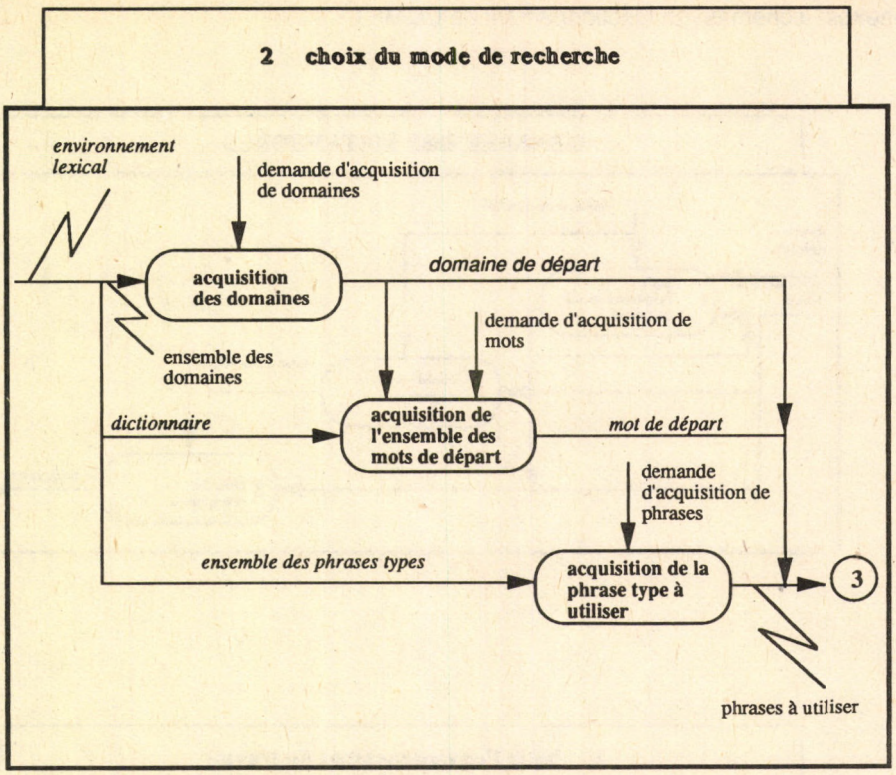
LAKOFF G. & JOHNSON M. (1980), *Metaphors we live by*, University of Chicago Press.

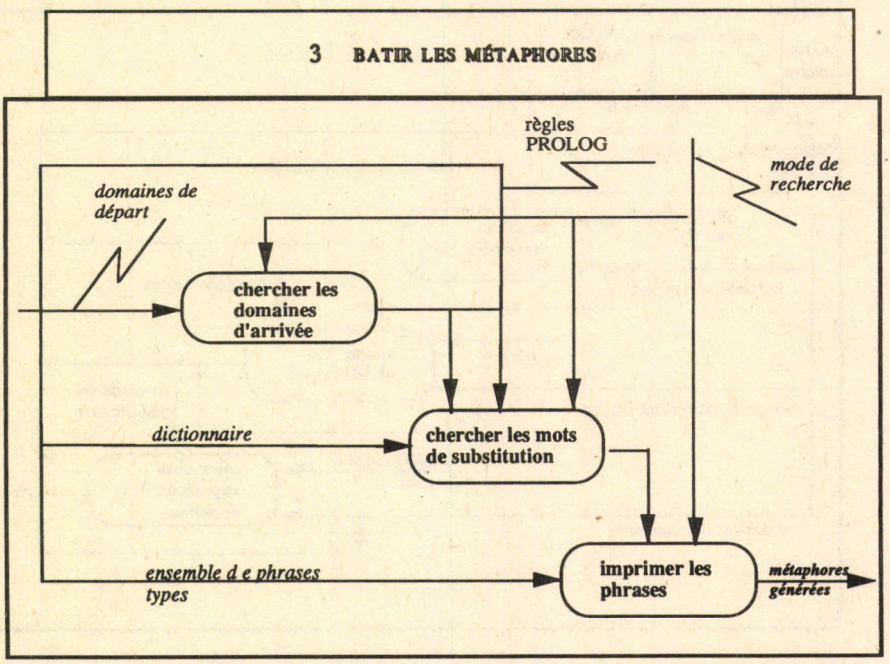
PROPP V. (1965), *Morphologie du conte*, Seuil (édition originale en russe : Akademia, Léningrad, 1928).

annexes: schémas de fonctionnement de COMET






2 choix du mode de recherche
environnement lexical
demande d'acquisition de domaines
acquisition des domaines
domaine de départ
demande d'acquisition de mots
ensemble des domaines
dictionnaire
acquisition de l'ensemble des mots de départ
mot de départ
demande d'acquisition de phrases
ensemble des phrases types
acquisition de la phrase type à utiliser
3
phrases à utiliser


3 BATIR LES MÉTAPHORES
règles PROLOG
mode de recherche
domaines de départ
chercher les domaines d'arrivée
dictionnaire
chercher les mots de substitution
ensemble d e phrases types
imprimer les phrases
métaphores générées

"PILAF": Software Tools for Lexicography and Linguistic Research

JACQUES COURTIN — DANIÈLE DUJARDIN —
IRÈNE KOWARSKI

ABSTRACT :

We present the PILAF system, composed of morphological and syntactic parsers, a morphological generator and interactive editors for all the linguistic data. These tools have been integrated in a user-friendly program, on Macintosh and PC. The system has been put to use for various applications : large-scale French language, written and spoken, small-scale models for other languages, several lemmatisers, error detection and correction in written French.

1 - THE PILAF SYSTEM

Among the research work effected by the **TRILAN** "**TR**aitement **I**nformatique de la **LA**ngue **N**aturelle" team [COURTIN 92], the **PILAF** project aims at building a linguistic toolbox, implemented on microcomputers . The tools it contains, tools which are necessary for interactive processing of natural language, may be put to use for different applications and by many different users. This adaptability implies that they must be not only general, parametrable and easily transported, but also that they should be easy to integrate in different systems.

At present, the **PILAF** system (**P**rocédures **I**nteractives **L**inguistiques **A**ppliquées au **F**rançais) is a user-friendly sytem for parsing the French language. Its general architecture is represented in figure 1.

The software proposes modules for morphological parsing, generation of forms associated with a root, and construction of dependency trees. It relies on a data base composed of two dictionaries and linguistic data including a validation-saturation grammar defined by a set of rules, as well as lists of models, of lexical categories and of variables . Furthermore, a set of dependency rules is provided in view of a syntactic parsing. All of this data is manipulated by means of specialized editors.

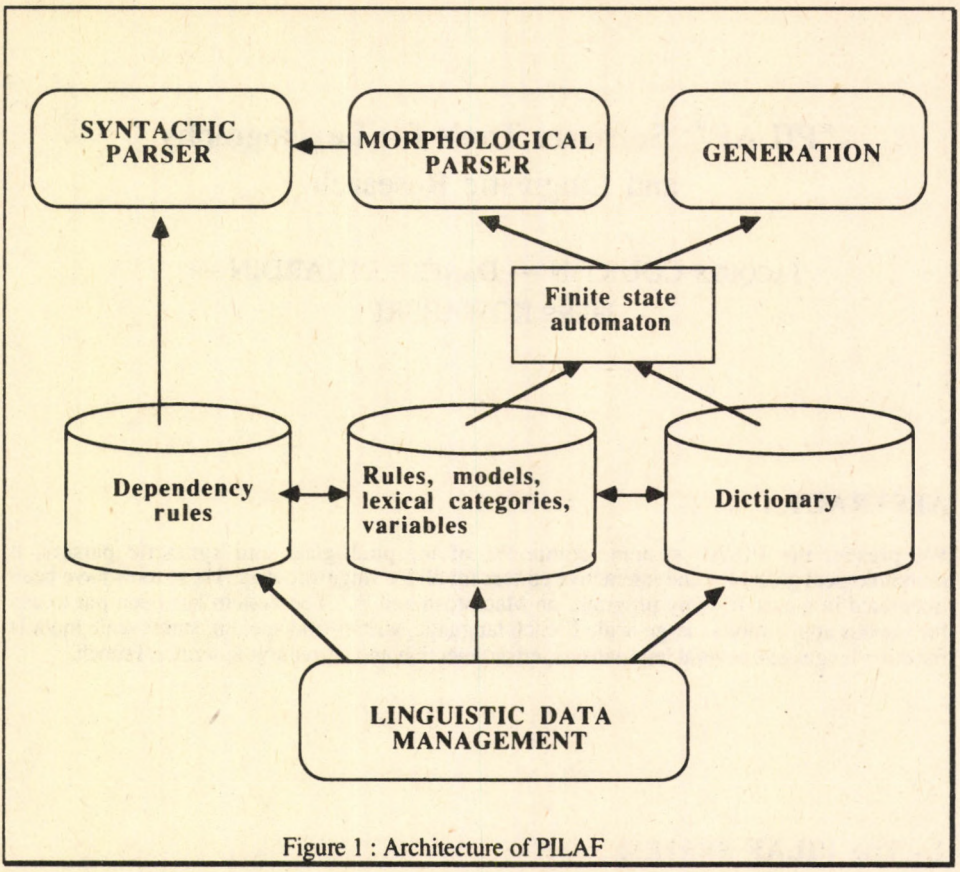


Figure 1 : Architecture of PILAF

Morphological and syntactic parsing are effected by use of a reversible finite state transducer which aims at :
- segmenting a character string in order to obtain the substrings which are components of it ;
- associating to each of these strings a certain amount of linguistic information.
In order to do this, the dictionary leads to determining a contiguous part of the entry string which is processed by the grammar in order to decide whether the concatenation of these different elements is or is not a valid. The grammar is a "Validation-Saturation grammar" (GVS) which is simply a more concise formulation of a finite state grammar.

THE MORPHOLOGICAL PARSER

computes, for any substring recognised in an entered character-string : a base, a lexical category, and grammatical variables.
Example : "The birds sing ." will give :

the	the	determiner	
birds	bird	common noun	animate, plural
sing	sing	verb	present, not third person singular
sing	sing	infinitive	
.	.	period	

Conversely, the use of **generation** gives all the flexional forms which derive from any base or any word, indicating for each its lexical category and the corresponding grammatical variables. Naturally, this list may be "filtered" in order to retain only some categories or variables.
Exemple : if the entry-word is `looks`, filtered by the categories : `verb` and `past-participle`, generation will deliver : `look, looks and looked.`

THE SYNTACTIC PARSER

uses a set of dependency rules in order to produce dependency trees such as :

```
          +-the °det
          I
       +-birds °cnn  /ani plu
       I
    =>+-sing °verb  /pre nth
       I
       +-in °advp  /
          I
          I   +-the °det
          I   I
          +-tree °cnn  /ina sin
```

2 -DESCRIPTION OF THE LINGUISTIC INFORMATION

THE LEXICAL CLASSES

According to Grévisse, French words are traditionally classified in nine categories or parts of speech : the noun, the article, the adjective, the pronoun, the verb, the adverb, the preposition, the conjunction and the interjection.
From this classification derives the idea of lexical category of a word. According to the type of application lexical classes may be split or may be grouped.

THE VARIABLES

Their aim is refinement of the results of morphological parsing. We can define not only grammatical variables such as singular or plural for number, but also variables which describe the function of the word in the sentence or any other information.

THE MODELS

ensure interfacing between the dictionary and the grammar. They are paradigms, or representatives of classes of words which have identical morphological behaviour (French verbs of the first group of conjugation for instance). Their form is the following :

/name/: - a list of names of rules applicable to this model ;
- various linguistic data;
- a list of names of rules applicable afterwards ;
- a list of names of rules forbidden afterwards.

The linguistic data provided for each rule and model can include :
- the lexical class of the word (ex: noun)
- variables (for example : feminine plural).

The choice of models will always appear to be arbitrary. Not only do they depend on the application : at least one model must be created for each lexical class, but some may be created as representatives of several lexical classes.

Example :the model **look** which delivers after morphological parsing the two homographs :
look common noun
look verb

THE GRAMMAR RULES

The sucession of grammar rules and their conditional application allow us to parse character strings and determine their linguistic information. Each rule represents in fact the analysis of an elementary string.
In this version of PILAF the possibility of prefixing which the transducer allows has not been retained.

The rules have the form :
/name /: - linguistic data;
- a list of names of rules applicable afterwards ;
- a list of names of rules forbidden afterwards;
- indicator of terminal rule .

THE DEPENDENCY RULES

They indicate, for any couple of lexical categories (gouvernor and dependant) their respective possible positions in the sentence, by means of a "list of weights".
Example : cnn * adj ::= -40 -20 16 would indicate that a common noun may be preceded by two adjectives , and may be followed by only one. The values of the weights are adjusted in order to allow or forbid insertion of other words between these two words.

UPDATING LINGUISTIC DATA

No linguistic data is fixed, the user is absolutely free to create and/or modify any category, variable, rule or model, and any entry in the dictionary, if the existing data does not suit him.

3 - THE TOOLS

The PILAF system described above is a composed of a certain number of functional modules, which were implemented until recently by means of independant programs, although they naturally used the same data files.

The main programs were : the morphological parser PILMOR, the syntactic parser PILSYN, the form generator PILGEN, as well as the dictionary editor PILDIC and the linguistic parameter editor PILPAR. Transferring from one program to another was frequently necessary,

and this was felt to be very inconvenient. For example, while parsing a text, the user noted that a given word was not in the dictionary. He would then have to leave PILMOR, enter PILDIC, insert the new word, leave PILDIC, and reopen PILMOR. Each time a program was opened, all the necessary files were reopened and reloaded in memory .

Another difficulty arose with the user-interface of each program which was fairly primitive : choice of an option in a menu by typing one character, very numerous confirmations of choices by typing yes or no, and particularly a very complex hierarchical tree structure which entailed numerous steps down and/or up in order to reach a function or move to another.

For these reasons we decided to make all the functions accessible by the shortest possible path starting from a single menu. The previous system had been implemented, programmed in Turbo Pascal, on PC compatibles. The Macintosh interface being particularly easy to implement, the first version of the new system was made by porting to the Macintosh (Think Pascal), and integrating the existing modules in a single menu. We then rewrote all the procedures for user interface and access to the functions of PILAF. We took the occasion of this transformation to integrate a few new functions, such as in-line help. Other functions, such as lemmatisation (particular case of generation) may easily be added. They call on the same database, the same transducer and the same elementary procedures.

This new system is at present being in turn readapted to PC compatibles, with improved ergonomy of the user interface.

4 - THE APPLICATIONS

The first applications of the PILAF system, and the most complete ones until now, are directed towards the French language.

WRITTEN FRENCH

[GROSS 86] [COURTIN 90] ,[COURTIN 91b]
In order to parse written texts, we now have at our disposal a grammar of 150 rules and 400 models, and a dictionary (enriched with the aid of the LADL electronic lexicon) with 35 000 bases allowing recognition and generation of 250 000 flexional forms.

Thanks to the ease of updating the dictionary and the linguistic parameters, we can parse texts as abstruse as the "dictées de Pivot" :
Aujourd'hui, je suis parfois si obsédée par la faim que, penchée sur les trésors de la Bibliothèque Nationale, je les confonds avec ceux de la gastronomie : manuscrits médiévaux et fricandeaux, palimpsestes minoens et courts-bouillons, in-folio et sot-l'y-laisse, ainsi que les culs-de-lampe historiés et les cancoillottes très parfumées, les incunables et les pets-de-nonne, les petits livres et les petits-beurre.
(DEMONSTRATION)

The first application of this version was detection and correction of spelling mistakes in French texts.[COHARD 88] ,[COURTIN 89b]
For detection, we have created, starting with the dictionary of bases and with the aid of the generator, a dictionary of flexional forms, sorted according to a "skeleton key". In order to correct simple typing mistakes, we propose forms which have tidentical or similar keys, and which derive from the erroneous form by one elementary transformation. We can also propose forms phonetically identical to the wrong form. Correction of grammatical mistakes : plurals, conjugations..., is obtained by means of parsing which delivers a base and grammatical variables, followed by generation which delivers a correct form.

Finally, we shall be able to detect and eventually correct concordancy mistakes by examining and modifiying the linguistic variables which decorate the dependency tree associated with a sentence [STRUBE DE LIMA 90], [COURTIN 91a] .

N.B. French spelling reforms :
More than 2000 words would be modified by the currently proposed reform, but the present spellings will remain acceptable and the new rules tend to limit exceptions.
Therefore, the new spellings can be parsed with the existing models. In some cases, we shall have to index new bases in the dictionary, when the base is changed from the present form.

SPOKEN FRENCH

[FRECHET 92]
We have studied spoken French and in particular spoken dialogue as a means of vocal command of a man-machine interface, in an attempt to define an efficient language for communication between a user and an application. For this research, users with different levels of computer competence were asked to comment freely and informally on their actions and intentions while manipulating software. The texts were recorded and then transcribed in written form, giving a corpus on which we studied the vocabulary (choice of lexicon and variety of syntactic patterns) by means of our tools for lexical parsing.

PILAF was chosen for its adaptability : not only was it possible to update the dictionary by adding the specific terms related to the domain of application (*to click, to zoom,...*) but we could also vary different parameters and adapt morphological classes to the specific needs of spoken language.

We have had to proceed to several adaptations entailed by the spoken aspect of the corpus and the specificity of the application, in particular the choice of a separator for spoken language. It can be noted that for written language, the system requires a separator such as the period, the comma or any other punctuation in order to limit a character string. For spoken language we chose the letter 'h' which signifies a respiration : it is the most common non-ambiguous sign in our texts.

Few locutions were indexed in the dictionary and according to our needs we had to input a certain number of them, such as these adverbial locutions : "par la suite", "en haut", "au milieu", "peu importe".

For certain fixed expressions we have sometimes had to create new lexical categories, (ex: "qu'est-ce que" interrogative word). It was often very difficult to know exactly which expressions we should retain and how they should be named because the study of spoken French shows the inapplicability of the traditional grammatical categories and conventional labels. For instance we have created a class called "speech support" ("appui du discours") containing words such as "alors", "donc", "ben", "e" and another called "pragmatic connector" which includes such expressions as "c'est fini", "je laisse tomber". We have included under the label "interjection" expressions such as "tiens donc", "pas de chance".

After having implemented all these adaptations, we were able to proceed to lexical parsing of our corpus.

The results

By using generation, we were able to associate to each form which appears in the text its *canonical form* or *lemma* : for any conjugated verb or participle, we deliver the infinitive, for a noun its form in the singular, for adjectives their masculine singular form...

lemma	form	lexical class	variables
je	j'	pper	sin fem mas
appeler	appelle	verb	sin tre uno pre ind
vocal	vocale	adjq	sin fem
cela	ça	œ	

We have therefore created three sorts of lists of lexical entities : lexical classes, lemmas, and flexional forms. For each list of lexical entities, the absolute and relative frequencies have been computed, then the lexical entities have been sorted by decreasing order of frequency in order to facilitate comparisons. By associating parameters to each lexical entity we were able to proceed to a systematic study of different aspects of the language such as the tenses of verbs.

By means of the creation of new lexical classes we were able to attain a more appropriate description of the problems of spoken language. We obtained the name and nature of each representative of each class for the whole corpus (20 141 items). We were also able to determine the degree of usefulness of each class for each user. In particular, we found that if the more expert users use more nouns, determiners and prepositions than the others, on the other hand they use less personal pronouns and verbs. Only 26 classes out of 51 were used by all the speakers .

Our knowledge of the nature of representatives, or models, of the classes we created ("speech support" and "pragmatic connector") helped us to proceed to a systematic study of intentionality.

We can note that lemmatisation, as well as production of statistics, are not part of the basic functions of the PILAF system, but are very easy to implement by making use of the functional modules of the system.

Example of a text belonging to our corpus :
*j'appelle le programme "PTS", à l'aide de: du clavier (h) et j'appuie sur "return", encore faut-il savoir que: (h) que ** que l(e) programme c'est "PTS" (=) (h) donc j(e) veux une "console graphique" "oui" et j(e) lance° par "return" j(e) veux une po la "position standard des fenêtres" "oui" (h) j'appuie sur "return°" bon j(e) laisse le: le fichier se: se mettre en place pour pouvoir choisir l(e) domaine où je veux aller (h),, (h) donc là: à partir de là (h) j(e) peux choisir e: (h) j(e) dois choisir le domaine d'étude (h) mais e: à partir de la souris puisque j(e) peux pas l'appeler par° commande vocale, choisis l(e) signal temporel puisque c'est c(e) qu'on m(e) fait étudier, alors il faut qu(e) j'appelle il faut qu(e) j'appuie sur le milieu de la souris mais e: ça il faut l(e) savoir aussi donc ça serait quand même plus facile de de parler, (+) et j(e) vais aller dans: "lecture" puisque j(e) vais étudier un: un son, dans l(e) "corpus public"*
(DEMONSTRATION)

LEMMATISERS

The Groupement d'Etudes pour la Traduction Automatique (GETA) in collaboration with TRILAN and using the tools provided by PILAF, has implemented a lemmatiser for French [TOMASINO 90]. This lemmatiser is integrated in a tool for indexing destined to serve as an interface for documentary bases. It is also used in the Malayan/French version of the SYSKEP system at the Sains Malaysia University of Penang.

We are also now undertaking the lemmatisation of the Phonetic Data Base BDPHO at the Institut de la Communication Parlée (ICP).

THE ENGLISH LANGUAGE

[ROBERT ET COLLINS 89] [WINOGRAD 83]
The English language has few rules for morphological flexion, therefore a small scale model for morphology (parsing and generation) was implemented very easily. On the other hand, its vocabulary is particularly extensive. We have limited our dictionary to a "mini-lexicon" of several hundred words, in particular the most frequent words, irregular verbs and "tool" words. .

Example of a text we process :
One of the students from Grenoble who started here in October, has decided to leave the course. She says she is depressed and cannot work. A major factor appears to be that she was with her grandmother during the Christmas vacation, when she was taken ill, and she subsequently died (the grandmother !). She has asked me to provide her with a certificate to say she was here from October until January. I presume this is connected with her grant from the departement. Is there anything else I need to do from the viewpoint of her grant ? If you have any contact with her it would be interesting to know if there are any other factors in her decision. Please give my regards to everyone, and I hope you enjoyed the Christmas pudding.
(DEMONSTRATION)

THE PORTUGUESE LANGUAGE

A small-scale prototype for processing the Portuguese language has been successfully implemented by Vera Strube de Lima [COURTIN 89a] , who is now pursuing her work on the subject at the Catholic University of Rio Grande do Sul (Porto Allegre, Brésil).

5- CONCLUSION

The PILAF programs are put at full disposal of all the university research teams who would like to use them.

Since last year [COURTIN 90], our software has been completely overhauled, rendered more convivial, with easier access. Our study on spoken French, with the corresponding updating of the database, is now finished.

Our more theoretical research on possible integration of semantic tools, on the level of the dictionaries as well as concerning parsing programs,[GENTHIAL 91],[GENTHIAL 92],is still under course.

REFERENCES :

[COHARD 88] : Brigitte COHARD, *Logiciel de détection et de correction des erreurs lexicales*. Mémoire CNAM, Grenoble, Mars 1988

[COURTIN 89a] : Jacques COURTIN, Danièle DUJARDIN, Irène KOWARSKI, Damien GENTHIAL, Vera Lúcia STRUBE DE LIMA, *Análise de textos escritos em português com PILAF - uma experiência e seus resultados*. 18avas Jornadas Argentinas de Informática e Investigación Operativa, Buenos Aires, Août 1989, pp 9.29-9.46.

[COURTIN 89b] : Jacques COURTIN, Danièle DUJARDIN, Irène KOWARSKI, Damien GENTHIAL, Vera Lúcia STRUBE DE LIMA, *Interactive Multi-Level Systems for Correction of Ill-Formed French Texts*. 2nd Scandinavian Conference on Artificial Intelligence, Tampere, Finland, June 1989, pp 912-920

[COURTIN 90] : Jacques COURTIN, Danièle DUJARDIN, Damien GENTHIAL, Irène KOWARSKI, *Creation And implementation on micro-computers of large scale French language dictionaries*. Conference on computational lexicography, Balatonszabadi , Hungary, September 1990

[COURTIN 91a] : Jacques COURTIN, Danièle DUJARDIN, Damien GENTHIAL, Irène KOWARSKI, Véra Lucia Strube de Lima, *Towards a complete detection/correction system*. International Conference on Current Issues in Computational Linguistics, Penang, Malaisie, June 91.

[COURTIN 91b] : Jacques COURTIN, Danièle DUJARDIN, *Paramètres linguistiques du français dans le système PILAF*. Rapport Technique RT 67, Laboratoire de Génie Informatique, Grenoble, Juin 91.

[COURTIN 92a] : Jacques COURTIN, Danièle DUJARDIN, Damien GENTHIAL, Irène KOWARSKI , *Outils lexicaux de l'équipe TRILAN : bilans et perspectives*. Séminaire Lexique du GRECO-PRC Communication Homme-Machine, Toulouse, Janvier 92

[FRECHET 92] : Anne-Lise FRECHET, Danièle DUJARDIN, Marie-Annick MOREL, Jean CAELEN , *Analyse lexicale pour le français oral avec le logiciel PILAF*. Séminaire Lexique du GRECO-PRC Communication Homme-Machine, Toulouse, Janvier 92

[GENTHIAL 91] : Damien GENTHIAL, *Contribution à la construction d'un système robuste d'analyse du français*. Thèse de l'université Joseph Fourier, Grenoble I, Janvier 1991

[GENTHIAL 92] : Damien GENTHIAL, Jacques COURTIN, *From Detection/Correction to Computer Aided Writing*. 14th CoLing, Nantes, July 92

[GROSS 86] Maurice GROSS, *Lexicon-grammar and the syntactic analysis of french*, 11th International Conference on Computational Linguistics, Bonn, Juillet 1986. p. 275-282.

[ROBERT ET COLLINS 89] *Dictionnaire Français-Anglais, Anglais-Français*, Dictionnaires Le Robert, Paris, 1989

[STRUBE DE LIMA 90] : Véra-Lucia STRUBE DE LIMA *Contribution à l'étude du traitement des erreurs au niveau lexico-syntaxique dans un texte écrit en français*. Thèse de l'Université Joseph Fourier, Grenoble I, Mars 1990

[TOMASINO 90] : Isabelle TOMASINO, *ODILE : Un Outil d'Intégration Extensible de Dictionnaires et de Lemmatiseurs*. Thèse CNAM, Grenoble, Décembre 90.

[WINOGRAD 83] Terry WINOGRAD, *Language as a cognitive Process*, Addison Wesley, 1983

A Bootstrapping strategy for Lemmatisation: Learning Through Examples

STEFANO FEDERICI — VITO PIRELLI

Abstract.

Automatic lemmatisation can be made slightly more exciting by the requirements imposed on the task domain by the use of large, unrestricted, textual corpora as a test-bed. Basically, the task remains the same, but the need of coping with a virtually open-ended linguistic material makes it rather more challenging. We would like to make the general point that such a need calls for "open-ended" automatic tools of analysis to be designed. In particular, NLP tools must be flexible enough to be able to automatically revise their own internal analysis procedures, and adjust them to new, unforeseen and some times even unexpected data. The line of research we sketch here draws on the notion of automatic, unsupervised learning through examples as a viable "bootstrapping" strategy for NLP. In this paper we illustrate the bootstrapping phase only, as carried out by a **parallel processing morphological system** developed in Pisa over the last few months.

Introduction

By **lemmatisation** we mean the process of associating a more or less fine grained morphological analysis to word-forms, which are presented one at a time as strings of characters. The input being taken out of context, the process of analysis cannot exploit syntactic constraints of any type. In case of morphologically ambiguous word-forms (homographs), more than one morphological interpretation must be provided as output, as illustrated by the following Italian example:

che relative pronoun (English "who/which")
che declarative conjunction (English "that")

Traditionally, pieces of software designed to handle lists of stems and inflectional paradigmatic tables of some kind, have proved to fare reasonably well in carrying out *lemmatisation* in this rather specific sense. Given a certain input word-form, they usually check off a list of available FORM / MORPHOLOGICAL_ANALYSIS pairs. Such a list is either precompiled, or generated at running time. If the input form is found in the list, the corresponding morphological interpretation which goes with it is given; if it is not, the analysis fails.

Automatic lemmatisation can be made slightly more exciting by the requirements imposed on the task domain by the use of large, unrestricted, textual corpora as a test-bed. Basically, the task remains the same, but the need for coping with a virtually open-ended linguistic material makes it rather more challenging.

We start the present work by giving an overview of what lemmatising usually implies. We then move on to cover more interesting grounds. We would like to make the general point that the need of coping with open-ended linguistic material calls for "open-ended" automatic tools of analysis to be designed. In particular, NLP tools must be flexible enough to be able to automatically revise their own internal analysis procedures, and adjust them to new, unforeseen and some times even unexpected data. This capability will be referred to later on as *self-modelling*. Experience in the field has shown that unrestricted data are often out of reach of the linguist's intuition, and therefore well beyond the predictive power of any rule-driven grammar whose nature and manner of operation are established once for all.

The line of research we sketch here draws on the notion of automatic, unsupervised learning through examples as a viable "bootstrapping" strategy for NLP. In the following we will concentrate on the bootstrapping phase only, as carried out by a **parallel processing morphological system** developed in Pisa over the last few months. More extensive testing is still under way. We believe that the sheer possibility of extracting a not negligible amount of extendable regularities from linguistic data is of theoretical interest *per se*. As far as the specifications of our system are concerned, here follows a list of the main points which will be touched on in the following sections:

a) how the system strips off affixes and assigns them appropriate morphological analyses, having no *a priori* knowledge of either affixes or lemmata;
b) how inflectional *paradigms* are consistently created;
c) how *paradigms* are extended to unknown words;
d) how suffixes do help appropriate analysis;
e) how multiple, potentially rival morphological analyses nicely mesh together in a parallel network to yield the expected result.

Corpus-driven linguistics urges for data-driven computational resources to be devised. What we offer here is still an embryonic but nevertheless promising case in point.

1. Lemmatising: minimal requirements.

What follows is the basic design of a lemmatiser whose intended usage is to provide, given a certain input word-form to be analyzed, the whole range of possible morphological analyses of the word-form in question in terms of:

a) the lemma(ta);
b) a bundle of selected morpho-syntactic features: namely part of speech, person, gender, number and tense.

To be more concrete, let us take a string like "porta", a typical example of an Italian ambiguous word-form. It can be either a feminine noun (English equivalent "door"), or a verb (English equivalent "brings/carries"). A lemmatiser is expected to provide this double analysis. In particular, our lemmatiser will yield, *inter alia*, an output of this type:

```
1)    "porta"=>          "porta/n f/n s/n"       (for    "door");
2)    "porta"=>          "portare/v 3/v s/v"     (for "brings").
```

Morphological features are expressed here in the compact format required by the input/output interface of the system. Note that "/n" and "/v" stand respectively for "noun" and "verb". "f/" is short for feminine, "s/" for singular, etc.. Given a certain grammatical category - say noun - features are always listed in the same order. The slash convention is justified by the necessity of distinguishing number/gender in nouns, adjectives and verbs. This is linguistically motivated by the fact that they do not share the same morpho-syntactic properties: in Italian *gender* is a **lexical** feature for nouns (meaning that it must be specified in the lexicon and is not generally predictable on the basis of other non-morphological properties of the lemma itself), while *gender* in adjectives is morpho-syntactically determined by the agreement with the nominal head it modifies in the sentence; as a result they cluster in different paradigm types (Matthews 1974 and 1992); a similar point can be made for *number* in verbs as opposed to the same feature in nouns. The slash convention allows us to keep them separate: e.g., *f/n* (feminine noun) is simply not the same as *f/a* (feminine adjective) since they are primitive units and cannot be further decomposed. This is a reasonable assumption as far as *learning* is concerned too, but we will not further pursue this point here.

2. Parallel processing: a bird's eye view.

Our learning system drew inspiration from neural network models as they have been implemented through parallel distributed processing strategies of different sorts (hereafter PDP) over the last few years (Rumelhart et al. 1986, Nakamura et al. 1990 among others). However, the approach we have followed here departs sufficiently radically from what has been traditionally done in the field. We claim that the relation between symbolic and sub-symbolic processing of linguistic data is much more complex than usually acknowledged so far in PDP circles. Nonetheless we endorse what we take to be the main thrust of PDP-like approaches, namely the powerful notion of parallelism over inferential operations, and the dynamic structuring of acquired knowledge in a self-monitoring and self-adjusting network.

Classical, rule-driven programs for NLP generally consist of a set of formal rules (logically if-then implications), sequentially interpreted and applied to input items. Rules of this sort are defined once for all and stored separately from input items. The operation of each rule proceeds independently of whatever other rules may exist.

Parallel models assume that information processing takes place through the interactions of a large number of simple processing units interconnected in *large networks*. Units can be conceived of as high-level equivalents of neurons in the brain, but the analogy is merely suggestive and arguably slightly misleading too (Smolensky 1988). Each processing unit notionally represents a prime at the level

of analysis the network is expected to perform [1].

The most important entities involved in PDP-processing are complex patterns of activity over many units. Each pattern consists of the simultaneous activation of mutually supporting units, and corresponds to a certain (partial) hypothesis/response of the system to the input/stimulus being processed (e.g., the identity of a certain string of characters). Each unit may participate in many such patterns (a certain character may occur in a certain position in many words). Therefore, more hypotheses can easily be conjured up and contemplated at the same time, through their being partially "turned on" by the same set of input units. This generates an unstable state of the system's network, which has to "cool off" for the system to reach a stable state and converge on one (or more) response(s) to the input at stake. Extracted patterns of regularities (as opposed to rules) have no implications singly. It is only the entire set of regularities that has any implications. Inference must be a cooperative process. The crucial role of such a cooperation comes out very clearly in a very simple case of the word-recognition task. Suppose we are faced with a string of characters having one character missing (say M_I_S_S_?_N_G), and suppose we want the system to guess which word it stands for. The straightforward paradox a sequentially operating logical system is faced with is that, in order to provide the missing letter, the system has to know first which word it is dealing with; however for it to pick up the right word in its list, it has to know first which character is missing. It is here that the simultaneous activation of multiple, rival, partially activated solutions in a parallel network comes in handy.

The building up of such a network of interconnected units is achieved through training: the system has to learn how to associate a certain response to a certain input, through being exposed to a number of correct stimulus/response pairs. Certainly, given the unsupervised nature of the learning process, the network can be skewed, and the system can be led astray. However, it is contended that adding further regularities can repeal conclusions that were formerly valid but that proved wrong at a later stage: in this respect, "parallel inference is fundamentally non-monotonic" (Smolensky *op.cit.*).

Last but not least, no principled line is drawn to separate rules from the items to which they apply. In fact, there is no question whether a given pattern of units should be stored directly in a repertoire of idiosyncracies (the Lexicon) or should rather be stocked in a set of more general statements about admissible input (the grammar). General patterns are elicited through a process of on-line generalization over particular patterns. More precisely, the process of generalization (regularity extraction) is the natural by-product of the storage-retrieval machinery of the system itself, which exploits the overall rate of associations/similarities among already learned patterns. In a sense, the network of interconnected units is a huge Lexicon itself which produces its own set of general statements by establishing excitatory/inhibitory links among partially idiosyncratic patterns (a similar model for a morphological Lexicon is suggested by Bybee 1988).

3. Some approaches to machine-learning modelling.

All approaches to the computational modelling of learning presuppose a certain amount of self-modelling. While in so-called **rule-based** approaches there is a rule-governed interface between tokens and their classification (see Pinker's

---

[1] For example, given the task of recognizing strings of characters as words, and given a certain input string of characters we want to identify as a word-form of a certain type, in a PDP-system the hypothesis about the identity of this word is distributed over a large number of primitive units being simultaneously activated. Such units are not words themselves (as it would be the case in a classical list-based approach), but rather characters, or, more appropriately if one wants the system to be able to 'read' input *with noise*, sub-characters: e.g., vertical, horizontal or slanting lines making up the block letters themselves (Rumelhart et al. *op.cit.*).

strategy below), **non rule-based** approaches bypass the rule interface, to describe the problem of knowledge extraction in terms of direct associative relations among the input items at stake.

Within non rule-based approaches, associative relations have been usually modelled either through so called "neural networks" (Rumelhart et al. *op.cit.*), or through estimated probabilities over state transitions in a transition network (e.g., Gilloux 1991), or through other probabilistic estimates of mutual associative relations (i.e., mutual information and the like, see for example Brown et al. 1990).

Within rule-based approaches, two strategies have been mostly attempted: a) learning is seen as a way to induce grammatical regularities either through elicitation from traditionally presented grammatical descriptions (e.g., Borin 1991) or by using statistical techniques on large text corpora (e.g., Atwell 1987); b) learning is a kind of grammar tuning, through a bootstrapping process, from an initial fairly general grammar, to an increasingly more specialised one (e.g., Briscoe and Carroll 1991).

In a third sense, machine learning has been used as a way to enrich lexical knowledge by relying on some form of prior grammatical knowledge (Brent 1991).

In the following, by way of illustration, we will hint at two typical examples of the non rule-based and rule-based paradigm.

Well known examples of **PDP learning models** are based on the stochastic distribution of weights over layers of **pre-wired** units in a network. At the end of rather lengthy drilling, such models have proved able to simulate "intelligent" behaviour: they can consistently provide expected output when exposed to either already encountered input, or brand new data.

The main feature of such models is that they can apparently dispense with any type of phrasal structuring, as familiar to the whole of linguistic theorizing (but see Fodor and Pylishyn 1988 for comments). We feel, however, that a learning model must be able to do something more than simply pattern pre-defined (sub-symbolic) units in a consistent way. The problem of how such units emerge through learning in the first place, whether they are somehow "innate", or rather tentatively posited in the process of making generalizations over known data, remains to be squarely faced.

Pinker's **candidate-hypothesization** model is an interesting example of a radically different, symbolic, cognitive tradition. Here, rule hypothesizing is modelled in terms of logical moves more familiar to a linguists' audience. Tentative, fairly constrained rules are put forward in the first place. They are then expected to become more and more general as new examples contribute to widen the range of their applicability (Pinker 1984 & 1988). As far as this sketchy overview is concerned, suffice it to say that Pinker's rules are elicited through the analysis of pairs such as *speak/spoke*, *ring/rang*, *get/got* etc.. Rules of vowel change are thus extracted and eventually merged:

"change: e -> o, class: g_t"

This rule turns "get" into "got" and reads: "change *e* into *o* in the context *g_t*" (see Wothke 1986 for a concrete example). However, this type of if-then context sensitive rules has a rather *ad hoc* ring. More powerful and simple principles should account for the overall mechanism of regularity extraction, so that the same comparatively small set of operations would govern various different aspects of learning in normative practices. The inferential operations we adopted and which are illustrated in the following, require comparatively little experience with the task domain to which they apply.

4. Our proposal.

In our architecture, training is achieved through feeding an already morphologically disambiguated text into the system. This marks an important difference from a number of other learning systems particularly aimed at extracting morphological information, which are rather spoon-fed with carefully chosen, paradigmatically arranged word-forms of a certain type.

Our system simply goes through a lemmatised text in which ambiguous word-

forms have been manually disambiguated. The system reads one word-form/morphological-feature pair at a time (whose format is repeated below for convenience) in the order in which it shows up in the text.

porta INPUT
porta/n f/n s/n OUTPUT

Most notably, each word-from in the text has been assigned one and only one morphological analysis, namely the correct analysis with respect to the particular context in which the word-form appears. Therefore, the potential ambiguity of some word-forms is something that the system must find out by itself. This happens when it stumbles upon a familiar word-form w which activates a certain system's response, but which, much to the system's dismay, is given a different analysis in the text. From that point in time on, the system learns that w can have two different sets of morphological features associated to it, and will systematically provide both of them every time w shows up again.

This strategy is very practical, since one can use as input any ordinary, already morphologically disambiguated text, and resembles the actual learning process of a child gradually going through a text, improving its knowledge as it goes on reading. We do not want to put too much emphasis on the psycholinguistic realism of this analogy though; the main point at stake here is simply that an effective bootstrapping strategy has to rely on a **minimal** set of predefined requirements on the expected input. Too many input requirements/constraints make the claim that a certain computational system actually simulates a learning session simply vacuous.

The actual training session of the system coping with some manually disambiguated text can be factored out into five steps:

i. reading of the form
ii. guessing of a possible analysis
iii. reading of the correct answer as provided in the text itself
iv. storing of it in its own network
v. whenever possible, restructuring of the patterns of the network according to the newly acquired information

Let us suppose that the system is faced with an INPUT/OUTPUT pair like the following:

(i)
credeva INPUT
credere/v 3/v s/v imp/v ind/v OUTPUT

where OUTPUT means: the INPUT string is the third singular form of the imperfect indicative of the (Italian) verb **credere** (English "(he) believed").

The core of the system is a self-expanding, parallel network. The fundamental conceptual entity of the system's network can be defined as follows:

PATTERN = (INPUT => OUTPUT)

where '=>' reads "linked with": fig.1 in the appendix shows an oversimplified case of PATTERN, where the top nodes are the INPUT nodes, while the bottom node is the OUTPUT node. Solid lines represent the associative links between INPUT and OUTPUT. INPUT nodes constitute the stimulus fed into the system, OUTPUT nodes its response to it. Actually the INPUT=>OUTPUT linking is rarely so direct as in fig.1. Usually, a fair amount of intermediate units go in between to make the relation accountable for a variety of different cases (e.g. paradigmatically conditioned allomorphy and the like). Such intermediate units will be referred to as HIDDEN UNITS.

Thanks to the network's parallel architecture, each pattern is made up out of a number of independently operating units (each circle in fig.1). Let us further suppose that the system is given the INPUT/OUTPUT (i) above for the first time. Accordingly, it creates the pattern in fig.1 in the network. As soon as a new pattern is discovered, the already existing patterns will independently be compared to it. Each of the INPUT units activated by the string *credeva* will, in

its turn, activate other string patterns (word-forms) where the same unit can be found. Not all such strings will be taken into account by the system in the process of revising the network's internal patterns, however, but only those, among them, which happen to be linked to OUTPUT nodes that overlap with the OUTPUT node of *credeva*. This way, the system looks for patterns of similarities between its old knowledge and the newly acquired one. If the search is successful newly induced patterns will be established.

Such a prose description can be given a more formal attire through the following semi-formulaic expression:

(1) OLD_PATTERN ≈ NEW_PATTERN ≡ NEWLY_INDUCED_PATTERN;

where '≈' reads *compared with*, and '≡' reads *yields*.

As we saw, PATTERN is a two-faced INPUT-OUTPUT entity; when two patterns are compared, the INPUT face of pattern_1 is matched against the INPUT face of pattern_2, and so are the OUTPUT faces as well. If any overlapping emerges on both levels (which is a pre-condition for this inductive routine to be triggered), we will have the following situation:

a) at the INPUT level, given STRING_1 and STRING_2:
i. string_OVERLAP = STRING_1 ∩ STRING_2
ii. string_LEFTOVER_1 = STRING_1 - string_OVERLAP
iii. string_LEFTOVER_2 = STRING_2 - string_OVERLAP

b) at the OUTPUT level, given MORPH_1 and MORPH_2:
i. morph_OVERLAP = MORPH_1 ∩ MORPH_2
ii. morph_LEFTOVER_1 = MORPH_1 - morph_OVERLAP
iii. morph_LEFTOVER_2 = MORPH_2 - morph_OVERLAP

where '∩' and '-' are the standard set-operators for **intersection** and **difference**.

The system thus creates:

(2) NEWLY_INDUCED_PATTERN-SET = {NEW1, NEW2, NEW3}

where '=' reads *consists of*, and

(3) NEW1 = (string_OVERLAP => morph_OVERLAP)
NEW2 = (string_LEFTOVER_1 => morph_LEFTOVER_1)
NEW3 = (string_LEFTOVER_2 => morph_LEFTOVER_2)

are the new patterns set up through the inductive routine. This way the network grows in complexity, while the relevant OUTPUT units emerge gradually through tentative inductive steps.

Equations above may look rather ad hoc. In fact, they have been originally formulated in a much broader context for a general-purpose learning system (see Federici 90). With a few changes, they have been used to design a learning tagger (see Federici & Pirrelli 1991b), and we are currently using them for a number of different NLP applications.

As a result of the application of the simple equations illustrated above, the system is able to cope with a comparatively wide range of different cases:

a) already encountered word-forms;
b) new word-forms whose lemma has already been encountered;
c) new word-forms whose inflectional-endings/derivational suffixes are already familiar to the system.

4.1. An illustrative example.

In this section we will take a glimpse at the dynamic functioning of the system through a simple series of consecutive learning sessions.

Let us assume the sytsem has already learned the following patterns:

PATTERN_1 = (*credeva* => **credere/v 3/v s/v imp/v ind/v**)

PATTERN_2 = (*temeva* => **credere/v 3/v s/v imp/v ind/v**)

Thanks to equations (2) and (3) above, the system will structure the internal configuration of dynamic units of fig.2 in appendix. The reader should note that now *credeva* is given the same analysis in two ways, since the pattern in fig.1 is not blotted out, but the new one in fig.2 is simply added to the network. As it will be shown later on, the system is well capable of coping with the existence of multiple patterns conveying the same morphological analysis with no problems of multiple equivalent responses. The coexistence of whole-word-form representations as in fig.1 (where available, that is for known tokens) and compositional representations as in fig.2 (induced by the system itself) in the network has an interesting psycholinguistic plausibility (Caramazza et al. 1985 and 1988).

A further new pattern is now fed into the system:

PATTERN_3 = (*credemmo* => **credere/v 1/v p/v perf/v ind/v**)

Again, the simple application of (2) and (3) will yield the overall network configuration illustrated by fig.3 in the appendix. Note that the system has now stripped off one more inflectional ending. As a result, it will output the appropriate analysis of the following previously unheard-of word-forms:

INPUT1= *tememmo* => OUTPUT1 = **temere/v 1/v p/v perf/v ind/v**
INPUT2= *volemmo* => OUTPUT2 = **?/v 1/v p/v perf/v ind/v**
INPUT3= *voleva* => OUTPUT3= **volere/v 3/v s/v imp/v ind/v**

where the question mark in OUTPUT2 stands for missing information: the system does not know of any lemma for *volemmo* at this stage yet. INPUT1 is an instantiation of case b) at the end of the previous section. INPUT2 falls into case c). INPUT3 is a combination of both b) and c): the system is now able to produce the entire morphological analysis of *voleva*, including the lemma, by means of the new information about INPUT2 acquired from the training text.

Clearly, erroneous guesses are always possible, since generalization through induction cannot be constrained *a priori*. Our experience, however, is that the system's built-in error-shooting and error-amending capabilities are smart enough for the success-rate of its performance to grow at a remarkably high rate (see performance-figures below), beyond doubt comparatively much faster than in ordinary stochastic approaches. What follows is a closer, but inevitably cursory look at such capabilities.

Whenever the system is exposed to a new word-form, all activateable patterns will be turned on to offer their response (their OUTPUT face). An unstable state of the system is thus produced. Eventually, the system will settle down onto the most plausible pattern (response), which will then be yielded as the winning output.

Rather informally, given the input word-form *i*, the most plausible pattern is *p* = (I => OUTPUT) such that *I* overlaps with *i* more extensively than any other activated 'rival' pattern does. Therefore, matching one particular pattern is not a matter of yes or no, but rather a matter of more or less. The current input may well activate a certain pattern only partially: the latter will win all the same, if there is no other pattern which gets a larger share of spreading activation. Partial activation is therefore of paramount importance in dynamic parallel networks, and a key to understanding the virtues of parallel processing in general. As far as the functioning of our system is concerned, the full formalization of this key-notion is slightly more awkward. In a nutshell, if *i* is the current input and *p* = (I => OUTPUT) has been activated by *i*, then $i \cap I$ is not empty. We call $i \cap I$ the *handle* of *p* with respect to *i*. The success of *p* as a winning pattern is then a function of the relation of *handle* with *I*. As we said, the bigger the *handle*, the more plausible is the OUTPUT associated with *I*. However if *handle* = $I < i$, then *p* can be beaten only by a p_i such that $handle_i = I_i > I$. We refer to this principle as **completing:**

a pattern is complete if every input unit in it is activated; complete patterns

always win over incomplete ones; if two alternative patterns are both **complete**, *the one containing more input units is the winner; if they are both* **incomplete**, *the winner has the higher ratio* **handle/I**

Clearly, if *handle* = *i* = *I* then the activated pattern *p* fully analyzes *i*. In sum, *I* is used to gauge a *system's confidence score* (more on it below) to be assigned to each system's analysis.

There is a further intriguing principle which is worth mentioning at this juncture to shed light on the issue of self-modelling. We do not want the system to question the analysis of already encountered items as acquired from the training text: explicitly learned word-forms must always override (newly) induced patterns, even if the latter fully analyze such word-forms. However, it would be far too *ad hoc* to achieve this result by stipulation. In our system the *once learned never forgotten* principle is the side-effect of a general, 'holistic' learning principle, to the effect that "denser patterns always override sparser ones" (Federici 90), where the density of a pattern is defined by the ratio:

number_of_links_from_INPUT / number_of_OUTPUT_nodes

Accordingly, the pattern of *credeva* in fig.1 is denser than in fig.2 (same number of links but one OUTPUT node more). To understand why the "holistic" principle is independently required for other reasons we have to look at some more features of our system.

Paradigms are set up through linking inflectional endings with the lemma from which they have been stripped off. In fig.3 in appendix the indirect paradigmatic link is clearly represented: the node *credere* is linked to both *-emmo* and *-eva*, which are, as a result, related to each other. Other types of links, namely inhibitory links, make sure that given a certain OUTPUT node - say 2/v p/v - and a certain verbal lemma, only the paradigmatically appropriate INPUT unit is activated, while the others get inhibited.

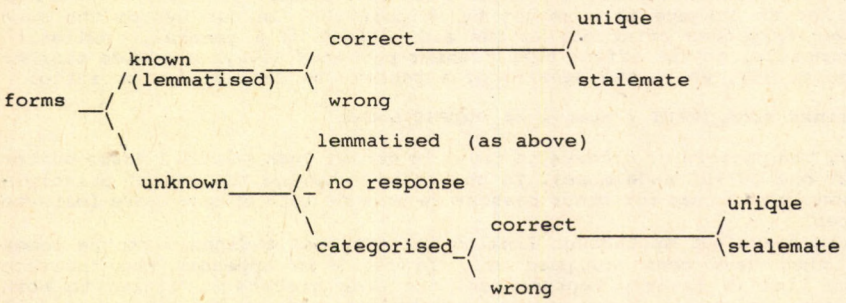
Paradigm extension is modelled through extending constellations of inflectional endings to other lemmata. This is based on the simple idea that once a given word takes a particular inflectional ending, it has to take all other paradigmatically related inflectional endings too (see how *temere* gets linked to *-emmo* in fig.3); otherwise it is an exception.

Exceptions are taken to be idiosyncratic patterns which cannot possibly fit larger, already established patterns of regularities. Accordingly, there is no principled divide between a regularity and an exception: the latter can also enter larger aggregates of exceptions, to end up giving rise to sub-regularities of some sort. Note that exceptions fail to behave compositionally; this implies that it is not possible for the system to further decompose the corresponding OUTPUT node into simpler nodes. A typical exception thus looks like the pattern of *credeva* in fig.1 in the appendix. Therefore, by the 'holistic principle', exceptions always override more regular patterns (which take more OUTPUT units). The reader should note that this is not achieved through ordering patterns from more specific down to more general ones (as in Kiparsky's *elsewhere condition* 1982, see also Wothke *op.cit.*), but it is again a by-product of the way the network operates, independently needed by other reasons.

5. Results of parallel bootstrapping.

To illustrate the performance of our bootstrapping routine, we report here some preliminary results coming from a comparatively small range test carried out on a SUN 3/50 using a prototype lemmatiser designed according to the principles sketched above. The test has been carried out this way: be *n* the number of word-forms already learned by the system by reading a disambiguated training text. After *n* word-forms being learned, the learning routine is switched off, and the performance of the system in analysis-mode only is reported. The column on the right hand side then reports figures concerning the performance of the system in reading **the following *n*** word-forms of a "raw" unanalysed version of the same text, and producing its own analysis. The analysis is evaluated by comparing the system's output with the disambiguated version of the same text. As hinted at

before, the system's response is always the pattern with the highest "confidence-score" among all activated patterns. If more than one response is given the same highest score, the system is in a stalemate, and offers more than one solution: e.g., this is the case when a morphologically ambiguous word-form is found which has been assigned two or more different analyses in previous occurrences. If the system's response does not contain the analysis annotated in the text, an error is counted. We have counted correct/wrong *unique responses* separately from correct/wrong *stalemate responses*. Moreover we have distinguished between *lemmatised word-forms* (where a full morphological analysis is provided, including the lemma) and *categorised word-forms* (where the lemma is not specified). In sum, the following hierarchy of possible results has been taken into account:



The second column on the right gives the overall percent amount. The third column on the right gives the percent amount with respect to Unknown Words. The reader should note that the curve of "correctly analyzed words" goes up steadily and steeply as the sample grows.

- n = 100

unknown word-forms	:	57	= 57.00 %	
(lemmatised) unique correct	:	0	= 0.00 %	
(lemmatised) stalemate correct	:	0	= 0.00 %	
(lemmatised) wrong	:	0	= 0.00 %	
(categorised) unique correct	:	4	= 4.00 %	
(categorised) stalemate correct	:	25	= 25.00 %	
(categorised) wrong	:	14	= 14.00 %	
no response	:	14	= 14.00 %	
correctly analysed	:	29	= 29.00 %	= 50.87 %
known word-forms	:	43	= 43.00 %	
(lemmatised) unique correct	:	27	= 27.00 %	
(lemmatised) stalemate correct	:	12	= 12.00 %	
(lemmatised) wrong	:	4	= 4.00 %	
correctly analysed word-forms	:	**68**	**= 68.00 %**	

- n = 1000

unknown word-forms	:	433	= 43.30 %	
(lemmatised) unique correct	:	28	= 2.80 %	
(lemmatised) stalemate correct	:	5	= 0.50 %	
(lemmatised) wrong	:	12	= 1.20 %	
(categorised) unique correct	:	195	= 19.50 %	
(categorised) stalemate correct	:	148	= 14.80 %	
(categorised) wrong	:	48	= 4.80 %	
no response	:	4	= 0.40 %	
correctly analysed	:	376	= 37.60 %	= 86.83 %
known word-forms	:	467	= 46.70 %	
(lemmatised) unique correct	:	404	= 40.40 %	
(lemmatised) stalemate correct	:	147	= 14.70 %	

(lemmatised) wrong	:	43	= 4.30 %
correctly analysed word-forms	:	**927**	**= 92.70 %**
- n = 2000			
unknown word-forms	:	670	= 33.50 %
(lemmatised) unique correct	:	92	= 4.60 %
(lemmatised) stalemate correct	:	16	= 0.80 %
(lemmatised) wrong	:	30	= 1.50 %
(categorised) unique correct	:	263	= 13.15 %
(categorised) stalemate correct	:	207	= 10.35 %
(categorised) wrong	:	76	= 3.80 %
no response	:	4	= 0.20 %
correctly analysed	:	578	= 28.90 % = 86.27 %
known word-forms	:	1330	= 66.50 %
(lemmatised) unique correct	:	949	= 47.45 %
(lemmatised) stalemate correct	:	335	= 16.75 %
(lemmatised) wrong	:	134	= 6.70 %
correctly analysed word-forms	:	**1862**	**= 93.10 %**

As illustrated below, the network's size grows gently, in spite of the powerful routine of paradigm extension. Some figures follow.

100			
order-dependent input units	:	323	= 323.00 %
hidden sample units	:	83	= 83.00 %
hidden suffix units	:	16	= 16.00 %
hidden lemma units	:	44	= 44.00 %
max link * lemma	:	1	= 1.00 %
max link * suffix	:	6	= 6.00 %
1000			
order-dependent input units	:	472	= 47.20 %
hidden sample units	:	502	= 50.20 %
hidden suffix units	:	187	= 18.70 %
hidden lemma units	:	781	= 78.10 %
max link * lemma	:	13	= 1.30 %
max link * suffix	:	69	= 6.90 %
2000			
order-dependent input units	:	500	= 25.00 %
hidden sample units	:	894	= 44.70 %
hidden suffix units	:	353	= 17.65 %
hidden lemma units	:	1740	= 87.00 %
max link * lemma	:	27	= 1.35 %
max link * suffix	:	240	= 12.00 %

LEGENDA:
hidden sample units express the number of different word-forms actually encountered in *n*; *max link * lemma* indicates the max number of suffix units a lemma unit is linked with; *max link * suffix* indicates the max number of lemma units a suffix unit is linked with.

Conclusions.
Note that the steepness of the success-rate curve is a crucial requirement of a bootstrapping strategy like tho one we advocate here. Further constraints over the set of admissible string-matching operations and on regularity-blending routines are similarly motivated. For a more flexible approach to learning see Federici & Pirrelli 1991a. All in all, we have shown that learning modelling through parallel processing does not necessarily imply lengthy, random, blind drilling. We are currently transferring the system's C code onto a Connection Machine at the Scuola Normale Superiore di Pisa. This seems to be an essential step if one wants to take full advantage of the fairly low (linear) complexity of the parallel algorithm.

References

Atwell, E. S. 1987 A Parsing Expert System which learns from Corpus Analysis, in W. Meijs (ed.) *Corpus Linguistics and Beyond* Amsterdam.

Borin, L. 1991 *The Automatic Induction of Morphological Regularities* Reports from Uppsala University Linguistics.

Brent, M.R., 1991, Automatic Acquisition of Subcategorization Frames from Untagged, Free-Text Corpora, in Proceedings of the 29th meeting of the ACL.

Briscoe, T., J. Carroll, 1991, Generalised Probabilistic LR Parsing of Natural Language (Corpora) with Unification-based Grammars, University of Cambridge, Technical Report No. 224.

Brown, P.F., J. Cocke, S.A. Della Pietra, V.J. Della Pietra, F. Jelinek, J.D. Lafferty,, R.L. Mercer, P.S.Roossin, 1990, A Statistical Approach to Machine Translation, *Computational Linguistics* 16:2.

Bybee, J.L. 1988 Morphology as Lexical Organization in M. Hammond, M. Noonan (eds.) *Theoretical Morphology. Approaches in Modern Linguistics* San Diego, Academic Press.

Caramazza, A. G. Miceli, M.C. Silveri, A. Laudanna, 1985 Reading Mechanisms and the organization of the Lexicon: Evidence from Acquired Dislexia *Cognitive Neuropsychology, 2* 81-114.

Caramazza, A., A. Laudanna, C. Romani, 1988 Lexical Access and Inflectional Morphology *Cognition 28* 297-332.

Federici S., (1990) Un sistema connessionista auto-espandibile di comprensione del Linguaggio Naturale Tesi di Laurea Universita' Pisa

Federici S. , Pirrelli V. (1991a) Doing Morphology without rules: an approach to linguistic knowledge acquisition through examples ILC-NLP-1991-2.

Federici S., Pirrelli V. (1991b) Tagger SECS: a Neural Environment for Corpus-driven unpublished.

Fodor J. A., Pylishyn Z. W. (1988) Connectionism and cognitive architecture :a Critical Analysis Cognition , vol.28 pp.3-71

Kiparsky, P. 1982 "Lexical Phonology and Morphology" Ms. MIT.

Matthews, P.H. 1974 *Morphology*, Cambridge University Press.

Matthews, P.H. 1992 *Morphology* (second edition) Cambridge University Press.

Nakamura, M., K. Maruyama, T. Kawabata, K. Shikano, 1990, Neural Network Approach to Word Category Prediction for English Texts, in Proceedings of COLING 90.

Pinker S., (1984) *Language learnability and Language development* Cambridge ,MA: Harvard University Press

Pinker S., Prince A. (1988) On Language and Connectionism: Analysis of a parallel distributed processing model of language Acquisition *Cognition* vol. 28 n.2 pp.2-193

Rumelhart D. E. et al. (1986) *Parallel Distributed Processing* MIT Press

Smolensky P., (1988) On the proper treatment of Connectionism *Behavioral and Brain Sciences 11*, pp.1-74

Wothke, K., 1986 Machine Learning of Morphological Rules by Generalization and Analogy *11th COLING* Bonn.

APPENDIX

fig. 1

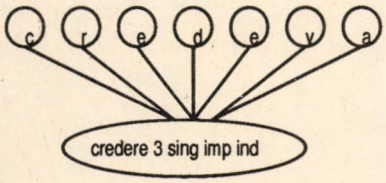


fig. 2

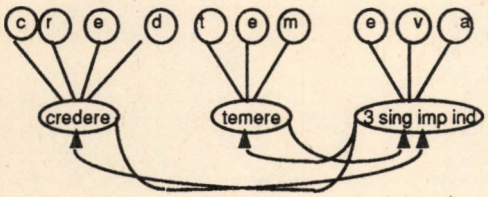
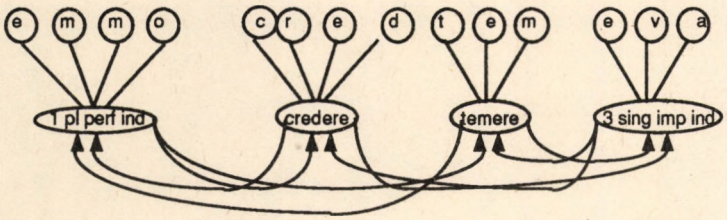


fig. 3



Co-occurrence Knowledge, Support Verbs and Machine Readable Dictionaries

THIERRY FONTENELLE

Introduction

Lexical acquisition has become a crucial research topic in computational linguistics and many researchers are convinced that it is not desirable to start coding thousands of lexical entries from scratch. Therefore the idea of re-using already-existing lexical resources (dictionaries or large textual corpora) has emerged. But using machine-readable dictionaries (MRDs) to feed the lexical component of NLP systems requires careful study of the microstructure of the dictionary and of the coding devices adopted by the lexicographers (cf. Boguraev 1991, Calzolari 1989, Fontenelle 1992a). In this paper, I wish to show that the machine-readable version of a bilingual dictionary, namely the **Robert & Collins English-French French-English dictionary** can be used to extract co-occurrence knowledge and information about support verbs. The results of the experiment described here are then compared to other methods advocating the use of statistical tools and large textual corpora to extract the same type of information (Smadja 1991, Church & Hanks 1990).

Co-occurrence relations

In a paper originally presented at the First Lexical Acquisition Workshop in Detroit, Smadja (1991) describes XTRACT, a co-occurrence compiler that retrieves lexical relations from a large statistically-analysed corpus. The lexical relations XTRACT acquires are co-occurrence relations, a.k.a. idiosyncratic collocations. This type of information is extremely important for language generation since it makes it possible to encode lexical constraints that account for the well- or ill-formedness of the following sentences:

(1) John quenched the fire.
(2) John extinguished the fire.
(3) John quenched his thirst.
(4) * John extinguished his thirst.
(5) John slaked his thirst.
(6) * John slaked the fire.

The term "collocation" refers to the syntagmatic combination of lexical items. These

constraints need to be encoded in order to avoid oddities in the generation process (cf. Smadja's examples: * *a powerful tea* instead of *a strong tea*; * *a strong car* instead of *a powerful car*). Since collocational restrictions are unpredictable, they need to be encoded either manually (by a team of lexicographers), which is both time-consuming and costly, or (semi-)automatically.

Smadja's XTRACT compiler extracts co-occurrence knowledge from very large textual corpora by identifying statistically relevant lexical relations. To quote Smadja's paper, "XTRACT takes as input a corpus d and a dictionary specifying parts of speech. It produces a list of tuples (w1, w2, cook-info), where (w1, w2) is a lexical relation between two open-class words (w1 and w2) identified in d, and cook-info is a set of statistical figures representing the lexical relations within the distribution of words collocating with w1" (e.g. decision_make_21.7657; decision_take_5.321). The following examples illustrate some results for two corpora consisting of approximately 2,300,000 words. Each row represents a productive word and its collocates are classified according to the part of speech (N=noun; V=verb; A=adjective).

law:	V: change, enforce, pass, violate
	N: court, universe
	A: Jewish, penal, physical
argument:	V: have, reject, settle, use
	N: court
	A: side, valid
food:	V: eat, have, prepare, sell
	N: health, industry, product, shortage, supply
	A: good, kosher, spicy, Yemenite

Smadja has then refined his algorithm to be able to specify that a noun used as an object of a verb is a direct or an indirect object (taking into account the distribution of words, the presence of relative clauses or passive constructions, etc). As can be seen above, this programme seems to be mainly efficient with respect to adjective-noun, noun-noun and verb-noun lexical relations. In what follows, I would like to demonstrate that this type of information can be enriched with co-occurrence knowledge extracted from the Robert & Collins dictionary.

Collocations in the Robert & Collins dictionary

The magnetic tapes of the Robert & Collins English-French French-English dictionary (Atkins & Duval 1978) were made available to our department for research purposes under contract with the publishers. The WordCruncher Text Retrieval Software package (running under MS-DOS) was chosen to exploit the content of the dictionary. WordCruncher has generated a general index (with the frequency of occurrence) of all the words that appear in the dictionary. The English words now appear in capital letters while the French words appear in small letters. The metalinguistic information that appears in italics in the printed version now appears between angled brackets (parts of speech, subject fields, co-occurrence information...).

A systematic approach has been adopted by the lexicographers to account for collocational constraints (in italics in the printed version, between angled brackets in our processed file):

- typical noun subjects of the verb headword appear in square brackets [];
- typical noun complements of the noun headword appear in square brackets [];
- typical noun objects of transitive verbs are unbracketed;
- typical noun complements of adjectives are unbracketed.

The following examples (from the English-French part) illustrate this approach:

abolish *vt practice, custom* supprimer; *death penalty* abolir; *law* abroger, abolir
do away with *vt fus* **(a)** *custom, law, document* supprimer; *building* démolir
forceful *adj person, character* énergique; *argument, reasoning* vigoureux, puissant
go through 1 *vi [law, bill]* passer, être voté; *[business deal]* être conclu, être fait, se faire
incontrovertible *adj fact* indéniable; *argument, explanation* irréfutable; *sign, proof* irrécusable
infringe 1 *vt obligation* contrevenir à; *law, rule* enfreindre, transgresser, contrevenir à
penal *adj law, clause* pénal; *offence* punissable
revocation *n [order, promise, edict]* révocation; *[law, bill]* abrogation; *[licence]* retrait; *[decision]* annulation
validate *vt claim, document* valider; *argument* prouver la justesse de

One of the major drawbacks of printed dictionaries is that they only provide users with one access path, namely the alphabetical order. The WordCruncher organisation of our MRD, on the contrary, enables us to instantaneously retrieve all the occurrences of a given word in italics, together with the headword under which it is found. A programme can then assign a syntactic/semantic link to the pair of collocates retrieved from the dictionary. This link is assigned automatically on the basis of typographical information combined with the part of speech of the headword. If we apply this programme to the examples above, focussing on the collocates of *law* and *argument*, the output is:

law:	adjective: penal
	subject_of: go through
	object_of: abolish, do away with, infringe
	modifier_of_noun: revocation
argument:	adjective: forceful, incontrovertible
	object_of: validate

If we start from the hypothesis that co-occurrence knowledge is spread across the entire MRD, we can retrieve words with their collocates and tag the pair of items with a label that accounts for the surface link between the members of the pairs. If we consider that 89 items co-occurring with *law* can be retrieved from the dictionary (+ 115 items collocating with *argument*; 216 items for *food*, etc), we can reasonably conclude that the information contained in the dictionary might complement the data obtained from corpus-based analyses (the transitive verbs that can take law as direct object are, according to Robert & Collins: *abolish, annul, carry out, circumvent, contravene, defy, disobey, do away with, elude, enact, enforce, establish, evade, get round, infringe, invoke, keep, neglect, obey, offend against, put into operation, override, promulgate, reform, repeal, rescind, respect, revoke, sanction, stretch, subvert, trespass against, uphold* and *vote in*).

Although this paper focuses on the acquisition of English collocations, it goes without saying that the method which I suggest here can also be applied to the French-English part of the dictionary. Since French lexicography has not given birth to *commercially-*

available dictionaries that would provide as rich a source of lexical information as some English learner's dictionaries do, the possibility of extracting such knowledge for French from the Robert & Collins dictionary should certainly not be dismissed. It would for example reveal that the word "loi" (=law) can collocate with transitive verbs such as *abolir, adopter, appliquer, approuver, concocter, contrevenir à, édicter, éluder, invoquer, obéir à, faire opposition à, réformer, respecter, ressusciter, sanctionner, supprimer, toucher à, violer* or *voter.*

Polysemy in collocations

Analysis of the list of words occurring as typical objects/subjects/head nouns in italics reveals that they are most often used in their basic, prototypical meaning. There may be cases, however, where a given polysemous word is used by the lexicographer in its various senses. This accounts for the distinction we have to make in analysing the list of verbs associated with the French noun *prix* in the French-English part of the dictionary. *Prix* is indeed ambiguous and can refer to either Eng. *price* or *prize* and the verbs that collocate with this noun usually depend on its meaning. The list of verbs should therefore be split into two sublists (Fontenelle 1992b describes in greater detail the collocations of the English noun *price* extracted from the dictionary and from a large corpus):

$prix_1$ (=prize): attribuer, avoir, décerner, décrocher, donner, emporter, remporter...
$prix_2$ (=price): augmenter, baisser, débloquer, dégringoler, diminuer, s'effondrer, geler, grimper, majorer, plafonner, réduire... (sample lists)

This means that caution should be exercised in extracting collocations since word meaning obviously plays an important part in the use that is made of collocations. If co-occurrence knowledge can be acquired automatically from the dictionary, the semantic interpretation requires human intervention and the first step is the disambiguation of the base of the collocation. It can also be noted that this interpretation by a lexicographer is also required when collocations are extracted from corpora (cf. Smadja's comment in section 3 of his paper). The following sections sketch various approaches that can be adopted to do this semantic interpretation.

Lexical functions

Examining the list of lexical collocations for *law* reveals that the relationship between the base and the collocator is variable. *Enact, establish, promulgate* and *vote in* can be considered as near synonyms whereas *abolish, annul, repeal, rescind* and *revoke* all express the opposite meaning. The first set refers to what Benson *et al.* (1986) call CA collocations (i.e. verbs denoting creation and or activation). The second set of items refers to EN collocations (verbs meaning eradication and/or nullification).

Such differences can also be accounted for in the framework of Mel'čuk's Meaning-Text Theory (cf. Mel'čuk & Žholkovsky 1988, Steele 1990). The most important component of this theory is the Explanatory Combinatory Dictionary (ECD). This dictionary is called combinatory because it is intended to display the combinatorial properties of words (Apresyan *et al.* 1969). To do so, it resorts to a well-defined set of **lexical functions** that express a meaning relationship between a keyword and other words with which it frequently co-occurs. The typical example given in the MTT/ECD literature is

the **Magn** lexical function, which expresses the relationship between a phenomenon and the highest degree of this phenomenon. For example, *Magn (pain) = excruciating* means that the adjective *excruciating* has to be used to express a very intense pain (other items such as *gnawing*, *keen*, *searing*, *sharp* ... can also be used to convey this meaning). Mel'čuk has identified approximately 60 lexical functions, ranging from traditional lexical-semantic relations (Syn=synonym; Anti=antonym; Gener=hypernym) to lesser-known relations (Son = typical sound of - as in Son(dog)=bark; Liqu = liquidate/eliminate - as in Liqu(file) = delete, erase...).

The lexical collocations extracted from the Robert & Collins dictionary can also be analysed in terms of lexical functions. The CA collocations detailed above can be represented as follows in the ECD framework:

$CausFunc_0$ (law) = enact, establish, promulgate, vote in

where Caus is the causative operator expressing the creation of something and $Func_0$ refers to the semantically empty verb which takes the keyword as its subject (to enact a law = to cause a law to come into force).

The EN collocations would be represented in the following way:

$LiquFunc_0$ (law) = abolish, annul, repeal, rescind, revoke...

where Liqu expresses the nullification of something.

The Real function is used by Mel'čuk to express the fact that the deep-syntactic actants comply with the requirement of the argument of the lexical function.

Real (law) = carry out, obey, respect

The main problem is that the linguist analysing the pairs of collocates has to identify the lexical function which relates the two items. In the ECD suggested by Mel'čuk, the lexicographer starts with a base (the keyword) and a list of lexical functions. The main task is to discover how a given LF is realized.

Support verbs

The type of data that can be extracted from the Robert & Collins makes it possible to enrich our description of the English lexicon with information about support verbs. Machonis (1991) defines support verbs as verbs that carry little semantic content and are used for syntactic support (cf. *commit suicide*, *make an analysis*). Gross (1981) analyses support verbs within the lexicon-grammar framework and notes that they embody semantic restrictions that are much more complex than selection restrictions. Many deverbal nouns usually need some kind of semantically empty verb which only conveys information about person, tense and aspect, as in:

(7) John brings forward an argument

John can be seen as the "subject" of *argument*. Support verbs can then be defined in terms of their property to preserve the relationship between the subject and the supported noun. This explains why *accept* cannot be considered as a support verb in:

(8) John accepted the argument.

Accept is not a semantically empty verb. It carries a special meaning together with information about tense, person and aspect. If we substitute *speech* for *argument* in (7), we end up with an unacceptable sentence:

(7') * John brought forward a speech

The reason is that the verb that can be used to support *speech* is *deliver*:

(7") John delivered a speech

Other verbs, such as *make*, *give*, *address* or *get out* can also be used as support verbs with respect to *speech*.

As can be seen above, support verbs actually represent a subset of lexical collocations. They will usually be found in the "object_of" class of verbs generated by our programme (In the above-mentioned example, four support verbs in our dictionary contain information about possible collocability with *argument*. John *brings forward*, *enforces*, *poses* or *puts forward* an argument).

It should be noted that support verbs roughly correspond to the type of lexical relation that can be encoded through the **Oper** lexical function used by Mel'čuk. $Oper_1$, $Oper_2$ ... refer to the semantically empty verb which takes the first, second ... actant of the keyword as its subject and the keyword as its direct object (Steele, 1990, p.57). The examples given in Steele's book are:

$Oper_1$ (attention) = pay
$Oper_2$ (attention) = attract

Analysis of the "object_of" class of *attention* in the Robert & Collins shows that it can be combined with many more verbs:

$Oper_1$ (attention) = concentrate, focus, turn
$Oper_2$ (attention) = arrest, capture, draw, engage, engross, excite, fix, invite, occupy, stir up, take up, win

Support verbs and machine translation

In a recent paper, Danlos & Samvelian (1992) suggest a methodology to use support verb information in machine translation. Their contention is that it is preferable to start from the predicative noun, which is the most informative element in a sentence, insofar as it is responsible for the selection of the accompanying verbs. The ultimate aim of such an approach is to avoid bilingual context-sensitive rules where the translation of a verb is determined by its object. Danlos & Samvelian (p.21) illustrate the traditional approach with French-English examples of such rules:

avoir (_ habitude) --> be in
perdre (_ habitude) --> get out of
prendre (_ habitude) --> get into

They suggest that these rules are unnecessary and that information at transfer level should be limited to a simple translation rule such as *habitude* --> *habit*. In the monolingual lexicon, however, each noun should be described in terms of the set of support verbs with which it can be associated. Each verb should also be assigned a given semantic feature reflecting the aspectual, diathetic or modal values conveyed by the combination of the two. Some of the semantic values suggested by Danlos & Samvelian (1992) are:

neuter: be in (_ habit)
inchoative (beginning of a process or state): get into (_ habit)
terminative (end of a process or state): get out of (_ habit)
durative (duration of a process or state): keep (_ ascendancy)
causative (diathetic value - causative denotation): give (_ hangover)

Instead of having numerous (and costly) context-sensitive rules at transfer level, the system would contain a richer monolingual description of lexical items. A noun such as *habit* would be described in the English lexicon as follows:

habit --> {*be in* (neuter), *get out of* (terminative), *get into* (inchoative)}

A similar entry would characterize *habitude* in the French monolingual lexicon:

habitude --> {*avoir* (neuter), *perdre* (terminative), *prendre* (inchoative)}

Danlos & Samvelian (1992) then explain how these types of information can be used in a transfer-based MT system such as EUROTRA. They also argue in favour of the automatic extraction of such collocational information from large corpora. The technique I have described above shows, however, that the Robert & Collins dictionary is, at least to some extent, a suitable candidate for extracting such information. Since it contains bilingual data, it is fairly easy to derive the relevant support verbs for English and French. The task of the lexicographer eventually boils down to assigning the correct semantic value to each support verb. Analysing the occurrences of *habit* in italics in the English-French part of the dictionary makes it possible to enrich the lexical entry described in the paper mentioned above. The monolingual lexical information for *habit-habitude* would be:

	habit	-->	*habitude*
inchoative	acquire, contract, develop, form, get into, take to		prendre, contracter
terminative	break off, drop, conquer, forsake, get rid of, give up, outgrow, relinquish, shake off, slough off, throw off		abandonner, se débarrasser de, se défaire de, perdre, renoncer à, rompre avec, surmonter, vaincre
causative	infix		inculquer
causative + terminative	wean (from)		détacher (de), détourner (de)

Again, it should be noted that the aspectual values of support verbs can also be represented in terms of lexical functions: **Incep** parallels the inchoative value, **Cont** refers to the durative aspect and **Fin** refers to the end of a process/state (terminative) - (see also Steele 1990, p.47).

Conclusion

The aim of this paper was to show that computational dictionary analysis can complement corpus-based lexicography in a useful way, since commercial dictionaries embody a lot of useful information that can be put to good use in a lexicographer's workbench. As noted by Smadja (1991), co-occurrence knowledge rarely exists in compiled form. A notable exception is the BBI dictionary - Benson *et al.* (1986) - which is only available in print and seeks to capture idiosyncratic collocations but which does not attempt to do the semantic interpretation the authors describe in the introduction to the dictionary. The type of approach I sketch in this paper should be seen as a contribution to lexical acquisition in order to pave the way for better collocational dictionaries usable in language generation and in machine translation.

The ultimate aim is also to enrich the monolingual lexica by adding a collocational dimension to the classical description of lexical items. We have seen that doing so can be economic in a machine translation approach since it greatly reduces the size of the transfer lexicon. Various formalisms, ranging from Gross's lexicon-grammars to Mel'čuk's Meaning-Text Theory can then be used to exploit the type of data extracted from a bilingual dictionary such as the Robert & Collins.

References

Apresyan Y, Mel'čuk I, Žolkovsky A (1969) Semantics and Lexicography: towards a new type of unilingual dictionary. In F.Kiefer (ed.) *Studies in Syntax and Semantics*. Reidel, Dordrecht-Holland, pp.1-33.

Atkins B.T. & Duval A. (1978) *Collins-Robert French-English English-French dictionary*. London & Glasgow: Collins; Paris: Dictionnaires Le Robert.

Benson M, Benson E & Ilson R (1986) *The BBI Combinatory Dictionary of English*. John Benjamins Publishing.

Boguraev B (1991) Building a Lexicon: The Contribution of Computers. *International Journal of Lexicography*, Vol.4, N°3, pp.227-260.

Calzolari N (1989) Lexical Databases and Textual Corpora: perspectives of integration for a Lexical Knowledge Base. *Proceedings of the First International Lexical Acquisition Workshop*, Detroit.

Church K & Hanks P (1990) Word Association Norms, Mutual Information and Lexicography. *Computational Linguistics*, Vol.16 (3), pp.22-29.

Danlos L & Samvelian P (1992) Translation of the Predicative Element of a Sentence: category switching, aspect and diathesis. *TMI-92: Proceedings of the Fourth International*

Conference on Theoretical and Methodological Issues in Machine Translation, Montreal, pp.21-34.

Fontenelle Th (1992a) Automatic extraction of lexical-semantic relations from dictionary definitions. *EURALEX'90 Proceedings - Fourth International Congress*, Biblograf, Barcelona, pp.89-103.

Fontenelle Th (1992b) Collocation acquisition from a corpus or from a dictionary: a comparison. *Proceedings of the EURALEX Fifth International Congress*, Tampere, Finland.

Gross M (1981) Les bases empiriques de la notion de prédicat sémantique. *Langages* 63, pp.7-52.

Machonis P (1991) The support verb 'make'. In F.Kiefer (ed.) *Computational Lexicography: Proceedings of the COMPLEX Conference*, Research Institute for Linguistics, Hungarian Academy of Sciences, pp.141-153.

Mel'čuk I & Žholkovsky A (1988) The Explanatory Combinatorial Dictionary. In M.Evens (ed.) *Relational Models of the Lexicon*, CUP, Cambridge.

Smadja F (1991) Macrocoding the Lexicon with Co-occurrence Knowledge. In U.Zernik (ed.) *Lexical Acquisition: Using On-Line Resources to Build a Lexicon*, Lawrence Erlbaum Associates, Hillsdale, New Jersey (in press).

Steele J (ed) (1990) *Meaning-Text Theory: Linguistics, Lexicography, and Implications*. University of Ottawa Press.

Dictionnaries éléctroniques des phrases figées: traitement d'un cas particulier: phrases figées — phrases à Vsup

AGGELIKI FOTOPOULOU

Abstract

In this paper we will try to discern the limits between fixed constructions and support verb constructions, after having defined these two terms. One observation triggered this paper:

the limits between support verbs constructions and fixed expressions are not always clear. One could even maintain that there is a visible scalar passage between these two types of sentences. We notice that certain phrases which have certain properties of support verb constructions also are fixed phrases. Let us give some examples from M. Gross (1981):

(1a) *Il y a de l'eau dans le gaz*

In the above example, the constituents cannot be changed, if the meaning of the sentence is to be preserved; however, they allow the presence of the operator - verb "mettre" :

(1b) *La venue de Max a mis de l'eau dans le gaz*

when the verb *être* is added :

(1c) **De l'eau est dans le gaz*

If the above is true (and we are going to present here some examples supporting this view and also try to offer criteria for their distinction) we are going to use it for the analysis and the codification of a phenomenon which is quite important for Greek.

1. Introduction

Le cadre théorique dans lequel nous nous plaçons est celui du lexique-grammaire. Dans ce cadre, qui est fondé sur la théorie transformationnelle de Z. S. Harris, c'est la phrase élémentaire qui est considérée comme l'unité de sens et non le mot. Ceci suppose la description systématique et la formalisation stricte des phrases de base de la langue, à partir desquelles peuvent être dérivées toutes les séquences appartenant à cette langue. Pour ce réaliser, il est indispensable de décrire avec précision la totalité presque de la langue. La précision de description signifie avant tout que le lexique du domaine à étudier doit être affecté le plus largement possible et qu'il doit être caractérisé en fonction de la syntaxe, c'est-à-dire des constructions et des relations de phrases dans lesquelles ce lexique est impliqué. Le domaine nous occupant ici est celui des phrases figées. Dans la perspective donc de l'élaboration du lexique-grammaire des phrases figées en grec moderne, nous avons recueilli, dans une première étape,

indifféremment, phrases figées et phrases à verbe support,[1] les limites entre ces deux types de phrases n'étant pas toujours bien distincts. Dans cet article, nous exposerons la procédure que nous avons suivie pour arriver à en réduire les zones de recouvrement.

Une forme est considérée comme figée lorsque au moins deux des éléments qui la composent ont une distribution unique ou très restreinte. Cette définition recouvre tout ce que nous appelons *locution* ou *mot composé*. Il peut donc s'agir de verbes composés (ou phrases figées), d'adjectifs, de noms ou d'adverbes composés. Notre attention ici portera sur les verbes composés. Et par conséquent, l'un des deux termes figés sera le verbe. Dans la phrase suivante :

N_0 VC_1 [2] = :

(1) *ο Νίκος έφαγε τα ψωμιά του*
Le Nikos a mangé les pains à lui (gén)
(Nikos a fait sa vie = est au seuil de la mort)

le sujet est libre (c'est à dire variable) mais la relation verbe - objet n'est pas modifiable. Outre ce critère de forme qui constitue pourtant le principal critère du repérage des phrases figées, des intuitions de sens interviennent également; le sens des mots ne permet pas d'interpréter leur combinaison.

Par *phrases à Vsupport*, nous entendons les constructions où la fonction prédicative est portée par le nom. Le terme *verbe support* désigne les verbes qui ne sont pas porteurs de sens mais qui supportent les marques de personne, de temps et d'aspect de la phrase. Ils peuvent être aussi supports de la nominalisation :

N_0 VN_1 = N_0 *Vsup Dét* N_1 *Prép* N_2

(1) *η Μαρία σέβεται τον πατέρα της*
La Maria respecte le père à elle (gén)
(Marie respecte son père)

= (2) *η Μαρία νοιώθει σεβασμό για τον πατέρα της*
La Maria ressent respect pour le père à elle (gén)
(Marie a du respect pour son père)

Les constructions à *Vsup* sont caractérisées par un ensemble de propriétés syntaxiques dont aucune n'est nécessaire et suffisante, mais prises ensemble, distinguent les emplois à *Vsup* des emplois ordinaires. Pour les propriétés des phrases à verbe support nous renvoyons à M. Gross (1981), G. Gross et R. Vivès (1986).

---

[1] Sur la notion du verbe support, voir Z. S. Harris (1964) et M. Gross (1981).

[2] Les notations employées ici sont de Z. S. Harris telles qu'elles ont été adaptées au LADL :

- les arguments syntaxiques des verbes portent des indices numériques: N_0 est le groupe nominal sujet, N_1 le premier complément, etc.
- par la notation *C* nous indiquons les termes figés,
- par *Vsup* est indiqué le verbe support,
- par *Dét* et *Prép* le déterminant et la préposition,
- par *Vmt* et *Vcaus* les verbes de mouvement et les verbes opérateurs causatifs de mouvement.

Dans les traductions littérales nous donnons certaines notations casuelles comme *gén* qui indique le génitif et *acc* qui indique l'accusatif.

Rappelons que les phrases à *Vsup* sont considérées comme des semi-figées ou plutôt comme un cas particulier des phrases figées (des formes *non rigides* ou *non opaques*) par un nombre important d'auteurs. Notons parmi les travaux respectifs celui de B. Fraser (1970) et plus récemment, ceux de N. Ruwet (1983) et de D. Gaatone (1992).

2. Critères de distinction entre phrases figées et phrases à Verbe support *(Vsup)*

La confusion entre phrases figées et phrases à *Vsup* s'installe souvent quand les phrases à *Vsup* présentent un certain nombre de propriétés qui les rapprochent des phrases figées, comme par exemple le figement des déterminants. La distinction entre ces deux types de phrases se complique également quand les variantes[3] des *Vsup* sont nombreuses (et pas encore listées) et souvent plus courantes que les phrases à *Vsup* élémentaire. Il y a donc des phrases que nous ne savons pas comment il faut les considérer : comme des phrases figées ou comme des formes dérivées de phrases à *Vsup*. Pour essayer de répondre on a eu recours à deux tests :

- le remplacement d'une phrase donnée, souvent une phrase à variante (lexicale ou aspectuelle de verbe support) ou une phrase à opérateur (causatif ou à lien[4]), par une phrase à *Vsup* élémentaire (*έχω/avoir, κάνω/faire, είμαι /être Prép*) ou à *Vsup* converse[5] comme *δίνω /donner*

---

[3] Les variantes des *Vsup* se divisent en deux groupes : le premier regroupe des variantes lexicales comme *πραγματοποιώ /réaliser* dans :

η Μαρία έκανε + πραγματοποίησε μία ενδιαφέρουσα έρευνα για τις θαλάσσιες χελώνες
La Maria a fait + a réalisé une intéressante recherche pour les tortues de mer
(Marie a fait + a réalisé une recherche intéressante sur les tortues de mer)

et le deuxième des variantes aspectuelles, c'est-à-dire des verbes qui sont porteurs de nuances aspectuelles par rapport au sens initial de la phrase :

η Μαρία (προέβη σε + τελείωσε) μία ενδιαφέρουσα έρευνα για τις θαλάσσιες χελώνες
La Maria (a procédé à + a terminé) une intéressante recherche pour les tortues de mer
(Marie (a procédé à + a terminé) une recherche intéressante sur les tortues de mer)

[4] L'application d'un opérateur causatif à une phrase simple *(Νίκος είναι σε δύσκολη θέση)* a pour effet d'ajouter un argument, le sujet :

η Μαρία έβαλε # ο Νίκος είναι σε δύσκολη θέση
La Maria a mis # le Nikos est à difficile situation

= *η Μαρία έβαλε το Νίκο σε δύσκολη θέση*
La Maria a mis le Nikos à difficile situation
(Marie a mis Nikos dans une situation difficile)

En revanche, l'opérateur à *lien* se distingue des causatifs par le fait qu'il se lie à un complément de la phrase à laquelle il s'applique sans augmenter le nombre de ses arguments :

η Μαρία έχει # ο Νίκος είναι με το μέρος της Μαρίας
La Maria a # le Nikos est du côté de la Maria (gén)

= *η Μαρία έχει το Νίκο με το μέρος της*
La Maria a le Nikos de la côté à elle (gén)
(Marie a Nikos de son côté)

[5] On définit ainsi les couples de phrases à *Vsup* comme *donner-recevoir* (en français); il s'agit des phrases dont les compléments sont permutés sans changement de sens.

- l'acceptabilité ou non du groupe nominal, formé en plaçant le sujet du *Vsup* comme complément de nom; (Rappelons que la formation du groupe nominal est une des propriétés essentielles des constructions à *Vsup*).

Pour appliquer ces deux tests nous construisons pour une phrase donnée une phrase à *Vsup* (éventuellement hypothétique); nous estimons d'une part son acceptabilité, et d'autre part sa relation avec la phrase de départ. Ensuite nous testons si la formation du groupe nominal est possible. Nous illustrons ce procédé par l'exemple suivant :

(1) *η Μαρία τρέφει (μεγάλη + βαθιά) εκτίμηση στον Νίκο*
La Maria nourrit (grande + profonde) estime à le Nikos
(Marie voue une grande estime à Nikos)

En première approximation, (1) pourrait figurer dans les phrases figées étant donné qu'elle est métaphorique et que la combinaison entre le verbe et le complément semble unique; mais la présence par ailleurs d'une phrase à *Vsup =: έχω/avoir* en relation avec (1) :

(2) *η Μαρία έχει (μεγάλη + βαθιά) εκτίμηση στον Νίκο*
La Maria a (grande + profonde) estime à le Nikos

et la formation du groupe nominal :

η (μεγάλη + βαθιά) εκτίμηση της Μαρίας (προς το + στο) Νίκο με εκπλήσσει
La (grande + profonde) estime de la Maria (envers le + à le) Nikos me surprend

permettent de cataloguer (1) comme une variante lexicale d'une phrase à *Vsup*

Cette opération est nécessaire parce qu'en grec moderne les phrases de base, surtout celles en *έχω* et *είμαι Prép*, sont d'une acceptabilité douteuse donc difficiles à repérer. Les variantes (aspectuelles ou lexicales) sont beaucoup plus usitées que les formes considérées de base et souvent la parenté syntaxique superficielle de certaines constructions dérivées cache des analyses différentes. Par exemple, dans les phrases :

N_0 *V* N_1 *Prép* C_2 =:

(3) *ο Νίκος κρατάει τη Μαρία σε αβεβαιότητα*
Le Nikos tient la Maria à incertitude
(Nikos maintient Marie dans l'incertitude)

(4) *ο Νίκος κρατάει τη Μαρία σε απόσταση*
Le Nikos tient la Maria à distance
(Nikos maintient Marie à distance)

les relations entre les trois termes (N_0, N_1 et N_2) diffèrent. Dans (3), N_1 =: *Μαρία* est le sujet de N_2 =: *αβεβαιότητα* dans une phrase à *Vsup =: είμαι Prép*:

N_1 *είμαι Prép* N_2 =:

(3a) *η Μαρία είναι σε αβεβαιότητα*
La Maria est à incertitude
(Marie est dans l'incertitude)

Le verbe *κρατάω* de (3) est donc un opérateur causatif (statique ou duratif) sur la phrase figée en *είμαι Prép* Par contre, dans (4) le verbe *κρατάω* peut être considéré comme une variante lexicale du verbe *έχω /avoir*:

N_0 VN_1 *Prép* C_2 =:

(5) *ο Νίκος έχει τη Μαρία σε απόσταση*
Le Nikos a la Maria à distance
(Nikos maintient Marie à distance)

En effet, la phrase (4a) en *είμαι Prép* n'est pas acceptée :

N_1 *είμαι Prép* N_2 =:

(4a) **η Μαρία είναι σε απόσταση*
**La Maria est à distance*

Ainsi, nous gardons (4) et (5) dans les listes des phrases figées. Par la suite, nous appliquerons ces tests à une série des phrases.

2.1. Les phrases à verbe support *έχω /avoir*

Prenons l'exemple :

N_0 VN_1 *(gén)* =:

(6) *ο Νίκος χαίρει άκρας υγείας*
Le Nikos jouit extrême santé (gén)
(Nikos jouit d'une excellente santé)

Il n'autorise pas la phrase à *Vsup =: έχω* :

N_0 *έχω* N_1 *(accus)* =:

**ο Νίκος έχει άκρα υγεία*
**Le Nikos a santé (accus)*

lorsque le modifieur du nom *υγεία / santé* est *άκρα /extrême*. Ce modifieur entre dans une combinaison unique avec *χαίρω/jouir* et *υγεία/santé* dans la phrase (6) et empêche également la formation du groupe nominal : ** η άκρα υγεία του Νίκου /*l'extrême santé de Nikos* En revanche, après un changement de modifieur dans :

N_0 VN_1 *(gén)* =:

(7) *ο Νίκος χαίρει εξαιρετικής υγείας*
Le Nikos jouit excellente santé (gén)
(Nikos jouit d'une excellente santé)

la paraphrase avec le *Vsup =: έχω* devient possible :

N_0 *έχω* N_1 *(gén)* = :

ο Νίκος έχει εξαιρετική υγεία
Le Nikos a excellente santé
(Nikos a une excellente santé)

ainsi qu'un groupe nominal : *η εξαιρετική υγεία του Νίκου /l'excellente santé de Nikos* Nous considérons que la phrase (6) dans ces conditions est figée. Au contraire, la phrase (7) est une variante lexicale de la construction en *έχω* qui peut être décrite au moyen de la substitution simple (M. Gross, 1981) :

(A) *χαίρω (gén) = έχω (accus)*

La phrase :

(8) *τα όνειρα της Μαρίας πήραν σάρκα και οστά*
Les rêves de la Maria (gén) ont pris chair et os
(Les rêves de Marie ont pris corps)

est une phrase figée (pas de groupe nominal : **σάρκα και οστά των ονείρων της Μαρίας*, combinaison unique entre le verbe et le complément). Pas de phrase à *Vsup έχω* non plus :

**τα όνειρα της Μαρίας έχουν σάρκα και οστά*
**Les rêves de la Maria (gén) ont chair et os*

Par ailleurs, la phrase (9) comporte l'opérateur *δίνω /donner* appliqué à (8) :

(9) *το ταξίδι στην Ιταλία έδωσε σάρκα και οστά στα όνειρα της Μαρίας*
Le voyage en Italie a donné chair et os aux rêves de la Maria
(Le voyage en Italie a donné corps aux rêves de Marie)

Nous allons par conséquent conclure que les phrases (8) et (9) sont des phrases figées bien que la relation qui s'établisse entre-elles a été observée lors de l'étude des extensions aspectuelles des constructions à *Vsup = : έχω /avoir*[6]

La possibilité d'avoir une paraphrase à *Vsup* n'est pas toujours un critère suffisant pour décider du caractère figé d'une phrase. Dans :

(10) *η Μαρία καλλιεργεί (Ε + τις) ψευδαισθήσεις (του Νίκου + στο Νίκο)*
La Maria cultive (E + les) illusions (de Nikos (gén) + à le Nikos)
(Maria nourrit les illusions de Nikos)

on ne peut faire commuter ni le verbe ni N_1 = : *ψευδαισθήσεις /illusions*.

[6] Voir R. Vivès (1983) et A. Fotopoulou (1989).

**η Μαρία αναπτύσσει (Ε + τις) ψευδαισθήσεις (του Νίκου + στο Νίκο)*
**La Maria développe (E + les) illusions (de Nikos (gén) + à le Nikos)*

**η Μαρία καλλιεργεί (Ε + τις) ελπίδες (του Νίκου + στο Νίκο)*
**La Maria cultive (E + les) espoirs (de Nikos (gén) + à le Nikos)*

La combinaison unique entre *καλλιεργώ* et *ψευδαισθήσεις* est un argument pour placer cette phrase dans les figées. Néanmoins, nous remarquons qu'il existe en fait une relation entre N_1 et N_2 (avec une modification des déterminants de N_1) dans une phrase à *Vsup =:έχω/avoir* ainsi qu'avec ses extensions aspectuelles *διατηρώ/garder* (valeur durative)et *χάνω/perdre* (valeur terminative) :

ο Νίκος (έχει + διατηρεί + χάνει) (Ε + Poss-0) ψευδαισθήσεις
Le Nikos (a+ garde + perd) (E + Poss-0) illusions
(Nikos (a + garde + perd) ((des + ses) illusions)

En conclusion, soit ce cas peut figurer dans les classes de figés à cause de la spécificité de la relation entre le verbe et C_1, soit il peut être considéré comme une phrase construite autour du verbe causatif *καλλιεργώ* qui opère sur la phrase à *Vsup ο Νίκος έχει ψευδαισθήσεις* Autrement dit, la phrase (10) est un cas limite entre construction figée et construction à verbe support. Nous l'avons finalement placée dans les tables des phrases figées.

2.2. Les phrases à verbe support *κάνω /faire*

Les phrases à *Vsup =: κάνω* se repèrent beaucoup plus facilement et elles posent moins de problèmes de distinction que les phrases en *έχω* et *είμαι*. Considérons la phrase :

(11) *ο Νίκος εκπόνησε το σχέδιο για το νέο κολυμβητήριο*
Le Nikos a élaboré le projet pour la nouvelle piscine
(Nikos a élaboré le projet pour la nouvelle piscine)

La phrase en *κάνω* qui lui est associée :

(12) *ο Νίκος έκανε το σχέδιο για το νέο κολυμβητήριο*
Le Nikos a fait le projet pour la nouvelle piscine

ainsi que le groupe nominal :

το σχέδιο του Νίκου για το νέο κολυμβητήριο είναι εκπληκτικό
Le projet du Nikos (gén) pour la nouvelle piscine est surprenant

éliminent (11) des phrases figées. Nous la considérons comme une variante lexicale de (12).

2.3. Les phrases à verbe support *είμαι Prép /être Prép*

Rappelons brièvement la relation qui s'établit entre les constructions en *είμαι Prép* C[7] et leurs variantes aspectuelles (les verbes de mouvement) :

[7] Des constructions de ce type ont été étudiées pour le français par L. Danlos (1980).

N_1 *είμαι Prép* C_2 = N_1 *Vmt Prép* C_2

(13) *η Μαρία είναι στον έβδομο ουρανό*
La Maria est au septième ciel
(Marie est au septième ciel)

= (14) *η Μαρία ανέβηκε στον έβδομο ουρανό*
La Maria est montée au septième ciel

Nous signalons aussi les relations remarquées lors de l'application de certains opérateurs (causatifs de mouvement) à des phrases en *είμαι Prép C*:

N_0 *Vcmt* N_1 *Prép* C_2 = :

(15) *τα φιλιά του Νίκου ανέβασαν την Μαρία στον έβδομο ουρανό*
Les baisers du Nikos (gén)ont fait monter la Maria au septième ciel

Dans (14), le *Vmt=: ανεβαίνω/ monter* est une extension aspectuelle du verbe *είμαι Prép* et la phrase (15) est construite autour de l'opérateur causatif *ανεβάζω /faire monter*. Les phrases comme (13) et les phrases à *Vmt* comme (14) ne figurent pas dans les classes respectives des phrases figées puisque toutes les deux dérivent d'une phrase à *είμαι Prép* et que ces phrases ainsi que les constructions qui leur sont associées, seront étudiées séparement[8]. Ce qui nous intéresse ici, principalement, c'est de placer dans les tables[9] des phrases figées les constructions causatives comme :

N_0 *Vcmt* N_1 *Prép* C_2 = :

(16) *ο Νίκος έβγαλε τη Μαρία απο τη μέση*
Le Nikos a fait sortir la Maria du milieu
(Nikos a éliminé Marie)

et les constructions à verbes de mouvement comme :

N_1 *Vmt Prép* C_2 = :

(17) *η Μαρία (βγήκε + έφυγε) απο τη μέση*
La Maria (est sortie + est partie) du milieu
(Marie est éliminée)

lorsqu'on n'observe pas de construction associée en *είμαι Prép*:

*N_1 *είμαι Prép* C_2 = :

**η Μαρία είναι στη μέση*
**La Maria est au milieu*

et de représenter également dans des tables la relation établie entre (16) et (17). C'est pourquoi, nous avons ajouté une colonne qui est marquée positivement chaque fois que pour une phrase figée donnée de

---

[8] L'étude de ces constructions en grec moderne est en cours (R. Moustaki 1992).

[9] Les données ont été notées sous forme des tables. Chaque table correspond à une classe des structures. Sur chaque ligne de ces tables figurent les phrases figées; chaque colonne représente une propriété distributionnelle ou transformationnelle.

GCNP2 [10], par exemple, nous avons une construction à *Vmt* associée, sans structure en *είμαι Prep* sous-jacente et sans que la phrase perde son caractère figé. L'intérêt théorique de cette observation est que la présence des ces formes intermédiaires illustre le continuum entre les formes à *Vsup* et les formes figées, continuum qui existe aussi entre les formes libres et les formes figées. L'intérêt pratique de cette observation (pour l'élaboration d'un dictionnaire, par exemple) est qu'il faudra prendre en compte ces formes "intermédiaires" et les traiter, éventuellement, différemment des phrases à *είμαι Prép C*. Nous donnerons quelques exemples de différents couples *Vmt* (verbe de mouvement) - *Vcmt* (verbe operateur causatif de mouvement) associés :

(18) N_0 **ανεβάζω** /faire monter N_1 *Prép* C_2 = N_1 **ανεβαίνω** /monter *Prép* N_2 [11]

οι έπαινοι του Νίκου ανέβασαν τη Μαρία στην εκτίμηση του Άρη
Les éloges du Nikos (gén) ont fait monter la Maria à l'appréciation du Aris (gén)
(Les éloges de Nikos ont fait monter Marie dans l'estime d'Aris)

= **η Μαρία ανέβηκε στην εκτίμηση του Άρη**
La Maria est montée à l'appréciation du Aris (gén)
(Marie est montée dans l'estime d'Aris)

(19) N_0 **βγάζω** /faire sortir N_1 *Prép* C_2 = N_1 **βγαίνω** /sortir *Prép* C_2

η Μαρία έβγαλε τ'άπλυτα του Νίκου στη φάρα
La Maria a sorti le linge sale du Nikos (gén) en public
(Marie a déballé le linge sale de Nikos en public)

= **βγήκαν τ'άπλυτα του Νίκου στη φάρα**
Est sorti le linge sale du Nikos (gén) en public
(On a déballé le linge sale de Nikos en public)

(20) N_0 **φέρνω** /amener N_1 *Prép* C_2 = N_1 **έρχομαι** /venir *Prép* C_2

ο Νίκος έφερε τη Μαρία στο φιλότιμο
Le Nikos a amené la Maria au zèle
(Nikos a amené Marie à faire preuve de zèle)

= **η Μαρία ήρθε στο φιλότιμο**
La Maria est venue au zèle
(Marie est devenue zélée)

La relation est la même pour quelques verbes qui ont comme composant le verbe **φέρνω** [12] :

---

[10] Voir annexe.

[11] Dans certains cas, comme ceux qui illustrent les exemples ((18) et (19) nous avons des paires de verbes morphologiquement apparentés.

[12] Le verbe **φέρνω** quand il se compose avec certaines prépositions devient **φέρω**

η Μαρία επανέφερε το Νίκο στην πραγματικότητα
La Maria a ramené le Nikos à la réalité
(Marie a ramené Nikos à la réalité)

= ο Νίκος επανήρθε στην πραγματικότητα
Le Nikos est revenu à la réalité
(Nikos est revenu à la réalité)

3. Conclusion

Les exemples ci-dessus présentés donnent une idée des problèmes rencontrés dans cette étude sur la distinction entre les constructions figées et celles à *Vsup*. Nous devons pourtant noter que les phrases qui gardent les propriétés essentielles des constructions à *Vsup* tout en présentant un certain figement entre deux de leurs termes sont assez nombreuses et que parfois le classement de ces cas parmi les phrases figées ou parmi les phrases à *Vsup*, peut être arbitraire (exemple (10)). Cependant, la procédure que nous avons suivie nous a permis d'arriver à certaines conclusions instructives. Nous avons regroupé en deux catégories les cas que nous avons définitivement placés parmi les phrases figées. Il s'agit de :

(a) formes figées telles que δίνω σάρκα και οστά = παίρνω σάρκα και οστά (exemples (8) et (9)) qui suivent certaines règles syntaxiques des phrases à *Vsup* mais sans la forme de base.

(b) formes figées qui présentent des propriétés des phrases à είμαι *Prép C* (c'est-à-dire les variantes aspectuelles *Vmt - Vcmt*) mais sans la forme en είμαι.

Ces deux possibilités des constructions figées - supports sont aussi rencontrées dans d'autres langues européennes, comme il a été prouvé par une étude comparative entreprise dans le cadre du projet Eurotra (A. Fotopoulou, M. Gavrilidou au workshop Eurotra 1991).

Le traitement de reconnaissance de ces formes lors de l'analyse automatique suppose donc une étude distributionnelle, syntaxique et sémantique de chaque expression en particulier, afin d'en délimiter la zone fixe. Rappelons que la zone fixe d'une expression figée est la partie de l'expression qui admet un nombre fixe de mots simples, même si ces mots sont susceptibles de variations morphologiques[13]. Et lorsque une expression figée, d'après E. Laporte, se construit avec un verbe support, comme dans les phrases :

N_0 *être un bon à rien*

nous ne considérons pas que le verbe support fait partie de la zone fixe puisque ce verbe peut être effacé ou remplacé par un opérateur ou bien par une variante aspectuelle. Or, dans le cadre de phrases (a) et (b), les propriétés des constructions à *Vsup* qui ont survécu sont restreintes. Ainsi, pour représenter ces formes, la solution la moins couteuse semble être celle où on inclut aussi le verbe dans la zone fixe et on a donc deux phrases figées distinctes.

[13] Cf. E. Laporte (1988).

ANNEXE

Nous donnons ci-dessous la liste complete des 13 tables des phrases à compléments figés en grec moderne avec la structure de définition et un exemple pour chaque table :

(GCDETF) $N_0\ V(Ddef + Dind)\ C_1 =:$

η Μαρία ακολουθεί την πεπατημένη
La Maria suit la terre battue
(Marie ne sort pas des sentiers battus)

(GCPOSS) $N_0\ V(C\,Poss\text{-}0)_1 =:$

ο Νίκος μετράει τα λόγια του
Le Nikos compte les paroles à lui (gén)
(Nikos pèse ses mots)

(GCDET0) $N_0\ VC_1 =:$

η Μαρία πέει μεγάλα λόγια
La Maria dit grandes paroles
(Marie tient de grands discours)

(GC12) $N_0\ V(C + N)_1\ (C + N)_2 =:$

ο Νίκος πότισε τη Μαρία φαρμάκι
Le Nikos a arrosé la Maria poison
(Nikos a fait de la peine à Marie)

(GCP1) $N_0\ V\,Prép\,C_1 =:$

η Μαρία μιλάει στο βρόντο
La Maria parle au vide
(Marie parle dans le vide)

(GC1PN) $N_0\ VC_1\ Prép\ N_2 =:$

η Μαρία καρφώνει το βλέμμα της στο Νίκο
La Maria fixe le regard à elle (gén) à le Nikos
(Marie a le regard rivé sur Nikos)

(GCNP2) $N_0\ VN_1\ Prép\ C_2 =:$

η Μαρία άφησε το Νίκο πάνω σε τη χάικα
La Maria a laissé le Nikos dessus à la douceur
(Marie a abandonné Nikos sur son nuage)

(GC1P2) $N_0\ VC_1\ Prép\ C_2 =:$

ο Νίκος χάνει το έδαφος κάτω απο τα πόδια του
Le Nikos perd le sol dessous de les pieds à lui (gén)
(Nikos a perdu pied (psychologiquement))

(GCP1P2) $N_0\ V\ Prép\ C_1\ Prép\ C_2 =:$

η Μαρία γλύτωσε παρά τρίχα αποβέβαια θάνατο
La Maria s'est sauvée de poil de sûre mort
(Marie a échappé d'un cheveu à la mort)

(GCP2P3) $N_0\ V(C+N)_1\ Prép\ (C+N)_2\ Prép\ (C+N)_3 =:$

η Μαρία δίνει λαβή για σχόλια στους συναδέλφους της
La Maria donne prise pour commentaires aux collègues à elle (gén)
(Marie donne prise aux critiques de ses collègues)

(GCGN) $N_0\ V(C\ N(gén))_1 =:$

η Μαρία έχασε τα ίχνη του Νίκου
La Maria a perdu les traces du Nikos (gén)
(Marie a perdu la trace de Nikos)

(GCGPN) $N_0\ V(C(N(gén) + Prép\ N))_1 =:$

η Μαρία έσπασε το τσαμπουκά (του Νίκου + στο Νίκο)
La Maria a cassé le culot (du Nikos (gén) + à le Nikos (accus))
(Marie a rivé son clou à Nikos)

(GCPN) $N_0\ V\ Prép(C\ N(gén))_1 =:$

η Μαρία μπήκε στο ρουθούνι του Νίκου
La Maria est entrée au nez du Nikos (gén)
(Marie tape sur le système de Nikos)

REFERENCES

DANLOS, L., 1980, *Représentation d'informations linguistiques : constructions N être Prép X*, Thèse de 3e cycle, Université Paris VII.

DANLOS, L., 1988, "Les expressions figées construites avec le verbe support *être Prép*", *Langages* 90, Larousse, Paris.

FOTOPOULOU, A., 1989, "Etude comparative des extensions aspectuelles des verbes supports *avoir/έχω, être Prép/είμαι Prép, faire/ κάνω* en français et en grec moderne", Mémoires du CERIL 4, Paris VII, Paris.

FOTOPOULOU, A., 1992, *Une classification des phrases à compléments figés en grec moderne*, Thèse de Doctorat, Université Paris VIII.

FRAZER, B., 1970, "Idioms within a transformational grammar", *Foundations of Language*, vol. 6, no 1, pp. 22-42.

GAATONE, D., 1992, "Les locutions verbales et le passif", *Langages*, Larousse, Paris, à paraître.

GIRY-SCHNEIDER, J., 1978b, *Les nominalisations en français. L'opérateur faire dans le lexique*, Droz, Genève.

GROSS, G., 1989, *Les constructions converses du français*, Droz, Genève.

GROSS, M., 1981, "Les bases empiriques de la notion de prédicat sémantique", *Langages* 63, Larousse.

GROSS, M., 1982, "Une classification des phrases figées du français", *Revue Québecoise de Linguistique* 11:2, Presses de l' Université du Québec à Montréal, Montréal.

GROSS, M., 1988, "Les limites de la phrase figée", *Langages* 90, Larousse, Paris.

GROSS, M., 1992, *Grammaire transformationnelle du français : 4 - Syntaxe des phrases figées*, Cantilène,Paris, à paraître.

GUILLET, A., 1990, *Une classification des verbes transitifs locatifs*, Thèse d'Etat, LADL, Paris VII.

HARRIS, Z., 1964, "The elementary transformations", T.D.A.P., Université de Pennsylvanie, réimprimé dans *Papers in Structural and Transformational Linguistics*, Reidel, Dordrecht, 1970.

LAPORTE, E., 1988, "Reconnaissance des expressions figées lors de l'analyse automatique", *Langages* 90, Larousse, Paris.

RUWET, N., 1983, "Du bon usage des expressions idiomatiques dans l'argumentation en syntaxe générative", *Revue québécoise de linguistique* 13:1, Presses de l'Université du Québec à Montréal, Montréal.

VIVES, R., et GROSS, G., 1986, " Les constructions nominales et l' élaboration d'un lexique-grammaire", *Langue Française* 69, Larousse, Paris.

VIVES, R., 1983, *Avoir, prendre, perdre: constructions à verbe support et constructions aspectuelles*, thèse de 3e cycle, Université Paris-VIII et LADL.

Extracting Linguistic information from machine-readable versions of traditional dictionaries - a metalexicographic method and some tools

ULRICH HEID — MATTHIAS HEYN — OLIVER CHRIST

Abstract

This contribution discusses some methodological issues of the reuse of traditional dictionaries dealing in particular with the types of problems encountered, with approaches to the identification and "therapy" of these problems and with computational tools developed to support this type of reuse activity. Problems treated concern the analysis and interpretation of hierarchical structures of dictionary articles, missing or implicitly given information, polyfunctional (ambiguous) or synonymous value names, and the statistical relevance of the available information. The approach we use is based on metalexicographic studies of textual structures of dictionary articles. The tools are designed in string handling languages (`awk`, `sed`, etc.) to check the structure of articles from all over the dictionary by extracting relevant freely definable pieces of article structures. Although referring to the results obtained from a detailed analysis of the *Oxford Advanced Learner's Dictionary* (*OALD*, 3rd edition, electronic version), we emphasize more the methodological and tool-oriented aspects of this work than the individual facts about the example dictionary itself.

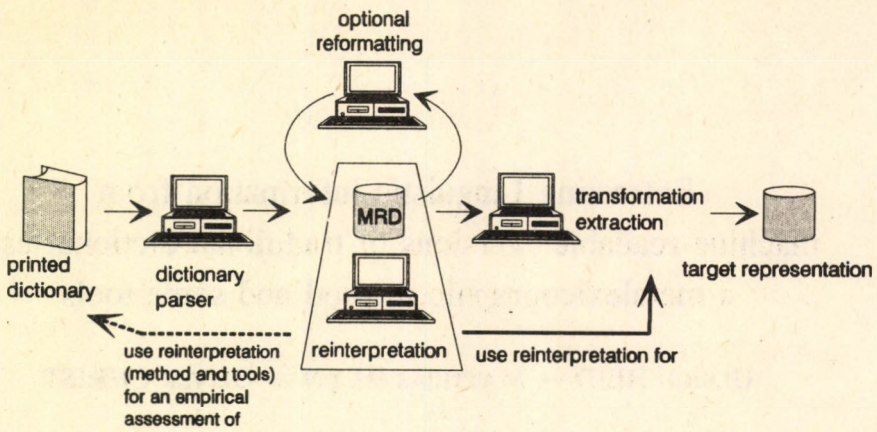


Figure 1: Working steps towards the reuse of MRDs

1 Goals and Framework

1.1 The problem

Most high-level Natural Language Processing (NLP) systems need large amounts of detailed linguistic information, and since dictionary construction is a time consuming activity, it is important to investigate possibilities of reusing electronically available versions of traditional dictionaries. In such a reuse situation, a paper dictionary would serve as a source of information, and the dictionary of the intended NLP system or the lexical requirements of the theoretical approach underlying the system would constitute the "expectation horizon" to be filled entirely or partly with material from the source. In the illustration in (1) we show the working steps towards the reuse of MRDs. The practical work necessary to extract linguistic information from an electronic version of a traditional dictionary involves a number of conceptually different steps which are often performed together. These have to do with *reformatting* the data available and with extracting relevant elements from there, wherever possible and useful. To this end, we have to gain an understanding of the presentational devices used in the dictionary, by *reinterpreting* the data. These processes, often performed at a time, can be a very time-consuming, iterative and error-prone enterprise. We thus consider it useful to do first "explorative" experiments before programming any computational tool for the reuse of a particular dictionary. The present paper is about such an "explorative" study and about the general methodology of such work.

Such a preliminary "explorative" study is most necessary if a *"traditional"* dictionary is considered as a potentially reusable source of linguistic information. We call "traditional dictionaries" those published dictionaries which have been produced according to the traditional lexicographic working procedures, without use of a computational system to enhance consistency, to facilitate the interaction among the authors of dictionary articles or to check whether the authors of articles stick to the guidelines given in the lexicographic instruction manual underlying their work. The third edition of the *Oxford Advanced Learner's Dictionary of Current English* is an example of such a "traditional dictionary". Since it is easily accessible for research purposes, we have used it for the work described in the present article[1]. However, for

[1]We would like to thank Oxford University Press for making available the electronic edition *(OALD3e)* to us for research purposes.

the argument of this paper, the use of this dictionary has much of an exemplary case-study; we do not wish to discuss in all detail the results obtained with this particular dictionary, but we are interested in the general types of problems posed by reuse of traditional dictionaries and in methods and tools for the exploration of such dictionaries.

The reuse of a traditional dictionary for the construction of an NLP lexicon involves, as we call it, *reformatting* and *reinterpretation*. By *reformatting*, we understand the process of translation from one representation format into another, say from a typesetting format to a representation of the dictionary as a text file with a markup[2]. In the case of OALD we only had to reformat from SGML to a notation as lists of LISP. This because conventional SGML-based tools could not be applied, since the OALD3e does not come with a document type definition for the tools to operate on. The *reformatting* process as such is nothing but an exchange of the encoding: it does not touch upon the signification of the dictionary text.
Reinterpretation, however, is the process of linguistic and lexicographic analysis of the contents of the dictionary text. Metalexicographers sometimes distinguish between the descriptive and the presentational side of lexicographic work. They thereby capture the fact that dictionary writing consists, roughly speaking, of two phases: a first one, description, where the dictionary author gathers material, structures and organizes it and describes the evidence he has before him, according to a given model; this model may be more or less formal(izable), of course; most often, it is encoded in the instruction manual. The second phase, *presentation*, aims at the writing of a dictionary article, which is a specific type of text, governed by rules; these rules ideally constitute the "syntax" or the "grammar" of the dictionary articles. Texts of dictionary articles are known to make use of numerous *presentational devices*, such as typographical change, abbreviations, punctuation, separator signs, sequencing of text elements, and so on. All these devices are meaningful, and so is the "wording" of the dictionary text: the dictionary author uses a number of devices to convey his descriptive intuition about the facts he has observed in the material he has gathered and analyzed. The task of reinterpretation is to understand the descriptive intuition of the dictionary author and to reconstruct, at least partially, the fragment of the linguistic facts of a language covered by his description. The reuse of a dictionary is the more easily and efficiently achievable the more we manage to relate elements of the textual presentation we find in dictionaries with observable linguistic phenomena or with our own way of formally describing these phenomena and of representing these formal descriptions. If some dictionary marks "transitive verbs" with a code "vt", it may be too naive to hope that we can safely assume that a formal grammar underlying an NLP system would necessarily classify the same set of lexical elements as "transitive verbs".

The goal of a reuse exercise may be to explore the dictionary as a whole and to reuse all information available, or at least as much as possible; we call this type of reuse activity *transformation*. Alternatively, only part of the information contained in a dictionary may be looked up automatically, e.g. to find evidence for a linguistic hypothesis or for a classification of certain types of facts; we call this type of reuse activity *extraction*. Extraction work is useful once we know which facts are most reliably and conveniently described in a given dictionary: we will then try to extract those.

1.2 The task

Our work is based on the *Oxford Advanced Learner's Dictionary*, *OALD*. The concrete task of reformatting and reinterpretation of the information contained in the *OALD* falls into two subtasks, each of which requires a certain set of methods and tools. The first subtask is to describe

[2]For example, this had to be done with LDOCE within the work done by the Cambridge group. See e.g. [Alshawi et al.1989].

in detail the internal text structure of the different types of articles found in this dictionary to investigate the availability, statistical relevance and descriptive adequacy of (certain types of) the linguistic descriptions presented to the user of this dictionary.

An anlysis of the internal structure of the articles is indispensable for a detailed and concrete enough description of the interdependencies of certain pieces of the article text; knowledge about these interdependencies is in turn needed for the formulation of extraction and transformation statements for a dictionary parser. The sequencing of different types of indications in a dictionary article text is not arbitrary, but may depend on a number of parameters, such as the type of the article, the presence or absence of certain other types of indications, etc. To be able to find the highest possible number of occurences of a certain type of indication, we have to know about the places within article structures, in which the respective indication can appear. The more we know about article structure, the higher the percentage of extractable indications with respect to those at all present in the dictionary. Knowledge about dictionary sturcture helps thus enhance the "recall" of the tools to be built for extraction or transformation.

Knowledge about accessibility, statistical relevance and descriptive adequacy of the actual linguistic description present in the dictionary is a basis for decisions about (selective) extraction of certain types of information: it is important to have an idea of the quantity (statistical relevance) and the quality (descriptive adequacy) of the potential extraction or transformation results. Any assessment of the accessibility of a given piece of information heavily depends on knowledge about the internal structure of the articles. Consequently, first article structure needs to be analyzed, before individual types of indications can be assessed.

1.3 The approach: methods and tools

The work on the reinterpretation and reformatting of the *OALD3e* combines *methods* from metalexicography, descriptive and theoretical linguistics with tools from "computational lexicographic practice": to arrive at describing in sufficient detail the internal structure of the articles of the *OALD3e*, we have adapted the metalexicographic dictionary description method of WIEGAND[3]. A number of computational *tools* have been developed to check both dictionary article structures (along the lines of our modification of WIEGAND's models) and the statistic relevance and quality of the linguistic descriptions presented in the *OALD*. This combination of a method for dictionary description with the tools supporting individual search and query tasks support the extraction of material from the dictionary for further manual analysis, e.g. with respect to descriptive adequacy. The resulting set of detailed descriptive statements about the textual structure of the dictionary articles, as well as about the properties of the presentational devices used in the dictionary, would give, together, a sophisticated transformation routine, allowing to re-represent the *OALD3e* in a well-defined homogeneous format.[4]

In the following section, we go through the most frequent types of situations occurring in the reinterpretation and reformatting of a traditional dictionary in view of its reuse in an NLP system. For each type of situation we indicate the problems to be solved and illustrate them with examples from the *OALD3*, then indicate the methods underlying the reinterpretation work and finally describe the functionality of the tools used to carry out the reinterpretation and reformatting exercise.

[3]The most recent comprehensive account of WIEGAND's methods for the description of dictionaries as texts can be found in [Wiegand 1991].

[4]We have carried out tentative transformations of the dictionary; one into an attribute-value notation as supported by unification grammars (represented in CLOS, Common Lisp Object System), and one into a marked-up text file format supported by a commercial look-up tool (MultiTerm). The first transformation did, however, not use the full set of the results of the structural and linguistic analysis. Its aim was to demonstrate the feasibility of such a transformation as such; cf. [Christ 1990].

2 Applying the methods and tools to concrete problems

Extraction and transformation routines for traditional dictionaries are usually very complex and highly specialized, because they have to keep track of numerous anomalies of dictionary articles. Such anomalies often have to do with the hierarchical structure of dictionary articles, with missing (but "expected") information, implicit statements, polyfunctionality or synonymy of value names or with idiosyncratic descriptions without statistical significance. Dictionary parsing presupposes ideally there to be a "grammar" for the textual structure of the dictionary articles (i.e. the order of functionally different text elements, interdependencies of text elements, etc. This "grammar" can for example be expressed by a documenttype definition (DTD) of SGML.). With traditional dictionaries, such a grammar often lacks or is not very detailed[5].

Rules governing text structure and interdependencies between pieces of the article text differ from dictionary to dictionary and have thus to be derived by induction, from the dictionary text. If these rules for article structures are not detailed enough, the dictionary parser will run into problems. However, the construction of rules for well-formed articles caters only for 85-90% of the articles. For modularity reasons, it makes sense to have a library of descriptions and rules for well-formed article structures and then to add a particular treatment of "exceptions" where appropriate. For setting up these rules we have based our work on WIEGAND's methods for the analysis of dictionary articles.

2.1 Article Structure: relating parts of article texts

2.1.1 Some examples

Dictionary articles consist of numerous text elements which each have different functions; these text elements are sometimes called *items*; WIEGAND's original term is *Angabe*. Individual such elements may have particular relationships and interdependencies with other elements. In particular, scope (or: validity) relations play a role in this context: we have to determine whether a statement about a linguistic property of a lexeme is valid throughout the article or overridden by a piece of information given later in the same article. This is most relevant in dictionaries with a complex microstructure, such as those which use *Nestbildung* in German, or *regroupement* in French lexicography or other devices for the integration of subentries into the main entries. Insufficient knowledge about the scope of individual (types of) items can lead to interpretation problems in hierarchically or recursively organized articles: the interpretation of article structures in dictionary parsing must allow to reconstruct the scope of each statement within a given article.

The following are some examples from *OALD*: In the article s.v. *archives*, the meaning description *"(place for keeping) public or government records; other historical records"* refers to the main lemma, *archives* reproduced in the illustration in (2), below, whereas the meaning description *"person in charge of ~"* refers to the sublemma *archivist*. The morphosyntactic indication *"n pl"* (which describes the main lemma as being a *plurale tantum*) has to be overridden by the morphosyntactic code *"n"* s.v. *archivist*, since with this word singular and plural

[5] Instruction manuals of dictionaries are a (more or less formal) description of the grammar of dictionary text. But these are usually not available for computational lexicographic or metalexicographic reinterpretation work; furthermore, they concentrate often more on the presentational aspect of the entries (e.g. typographic conventions, etc.) than on a definition of the types of phenomena to be described, e.g. doing linguistic tests for classificatory purposes. Often lexicographers do not follow in all detail the instruction manual, or it is not sophisticated enough to cater for the situations a lexicographer may encounter in practical work. Syntactic consistency of article structures will be ensured, in future, in those dictionaries which have been produced with the support of a computational system for structural control. The most well-known of these is the GESTORLEX system which has been produced by a Danish software house.

ar·chives /ˈɑːkaɪvz/ *n pl* (place for keeping) public or government records; other historical records. **archi·vist** /ˈɑːkɪvɪst/ *n* person in charge of ~.

Duce /ˈduːtʃeɪ/ *n* (I) leader (esp as used of Mussolini /ˌmʊsəˈliːnɪ/ (1883–1945) Italian Fascist leader).

Figure 2: Articles *archives* and *Duce*

are possible. The ~ takes up the lemma *archives* in the meaning description of the embedded derivative *archivist*, but otherwise the sublemma does not "inherit" any particular information from its main lemma.

A more specific (and anomalous) case are glossations within articles: in this case, the explanatory item (e.g. a morphosyntactic or a phonetic item) does not at all refer to a unit with lemma status: in the article s.v. *Duce*, given in the illustration in (2), the pronounciation item given within the semantic comment does not refer to the main lemma (*Duce*), but to an immediately preceding word of the semantic comment itself, the name *Mussolini*.

2.1.2 The method: path analysis

We distinguish two types of textual (and in part typographic, non-textual) items in dictionary text: one type conveys the actual linguistic description (e.g. a meaning description), the other is an indicator of the structure of an article. The latter type is more often non-textual (e.g. delimiter signs) and primarily allows to determine the scope of the items within a given article. Items which convey linguistic information can easily be represented, in the output of the transformation or extraction, in the form of attribute-value pairs as used in unification-based formalisms.

We describe both sorts of items in terms of their *item type* (function of an item, e.g. the type "meaning description") and of their *item value* (the actual textual form, e.g. the text of a meaning description).

Our analysis produces the following types of results:

- complex articles can be reduced, for further detailed analysis, to a few structurally relevant elements;
- an explicit description of textual interdependencies is given; this allows, for translation into an attribute-value format, to "collect" the relevant items for main lemmas and sublemmas, keeping track of inheritance and overriding in this process of collection of material;
- the recurrency of types of article structures can be checked throughout the dictionary and thereby the amount of "well-formed" and "ill-formed" articles can be identified;
- the results of the analysis serve as the basis of the specification of extraction and transformation rules.

Without going into all details, we now schematically describe the analysis of an article; taking as an example the article s.v. *archer* from the *OALD*[6].

[6] For a detailed description of the method, see [Heyn 1992].

archer /ˈɑːtʃə(r)/ *n* person who shoots with a bow and arrows. **arch·ery** /ˈɑːtʃərɪ/ *n* [U] (art of) shooting with a bow and arrows.

Figure 3: Article *archer*.

```
(ent :h "archer" (hwd "archer")(pr (ph "\"A:tS@(r)"))(hps :ps "n"
  (hsn (def "person who shoots with a bow and arrows")
       (cd (cp "arch|ery")(pr (ph "\"A:tS@rI"))
          (cps :ps "n" :cu "U"
             (csn (def "(art of) shooting with a bow and
                        arrows")))))))
```

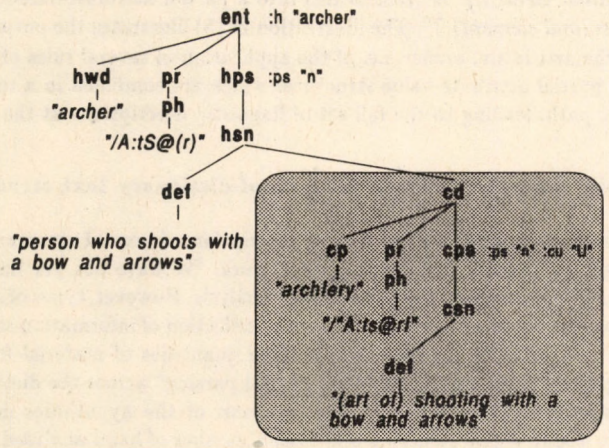


Figure 4: LISP coded version of the article *archer* and its hierarchical structure.

For practical reasons we have reformatted the SGML-like articles of the electronic edition into lists of LISP expressions. Illustration (3) reproduces the article as we find it in the OALD; the illustrations in (4) contains the representation of the article in a LISP expression, as well as a tree-like schema of its hierarchical structure.

This latter figure shows clearly the recursive nature of the article: the part of the article which is devoted to the derivative (*archery*, in the grey box in the scheme in (4)) has almost the same internal structure as the main article. This fact supports the transformation of the complex article, when reusing it for NLP purposes, into *two separate* simple articles[7].

Furthermore, the illustration shows clearly the two types of nodes representing the two types of items introduced at the beginning of this paragraph, namely *linguistic information indicator*

[7]This *flattening* is what has been done regularly in our transformations. Of course an explicit trace has to be kept of the fact that there has been a derivative entry embedded into another entry. The embedding is the presentational means used by the lexicographer to indicate that there is a derivational relation (affecting morpholgy, syntax and semantics of *archery*) between the derivative and the base. The same *flattening* of recursive dictionary structures is done in the ACQUILEX reformatting procedures, cfr. [Cal et. al 1990].

items and *structure indicator items*:

- *Information indicators*: items which convey linguistic information come as nodes and their terminals are linguistic "values"; cf. `hwd`, `cp`, `ph` or `def`. Alternatively, they have attribute-value pairs attached, such as in the case of `hps` or `cps`.
- *Sturcture indicators*: items indicating the textual structure of the article come as non-terminals with one or more daughters, but mostly without linguistic attributes, such as `cd`, `hsn` or `csn`.

We have analyzed in detail the structure and hierarchical organization of the the *OALD* articles in order to arrive at rules for transformation of the articles into an attribute-value pair based representation. For example, one rule says that a partial graph containing the sequence of nodes `ent - hwd` {*terminal element*} is transformed into a partial attribute-value-structure of the form `[form:` {*terminal element*} `]`[8]. The illustration in (5) illustrates the output of a complete path analysis of the article s.v. *archer*, i.e. of the application of several rules of the above type. It also shows the partial attribute-value-structures which are combined in a top-down manner along the relevant path, leading to the full set of linguistic descriptions at the bottom.

2.1.3 The tools: computer-assisted analysis of dictionary text structure

As a first step, the formulation of rules for the description of textual structures of dictionary articles has to rely on *manual* metalexicographic work. We have not yet implemented fully automatic transformation rules on the basis of path analysis. However, types of structure graphs and rules for automatic parsing of articles and for the collection of information along the relevant "paths" have to be constantly checked against large quantities of material from all over the dictionary. This process (a sort of "explorative partial parsing" across the dictionary) can only be carried out with computational tools. On the basis of the hypotheses derived from our general knowledge about article structure and from a number of hand-analysed sample articles, string handling programs are implemented in `awk`, `sed` and similar languages and applied to all articles of the dictionary. The results of the application of these programs allow for further refinement and modification. The programs are mostly parametric: types of structures, denoted by a list-like expression of embedded structure and information indicator items, are specified, and the tool extracts from the whole dictionary all structures which follow the model specified. One of the advantages of theese string handling languages is that they support a prototyping-like procedure. Programs can easily be refined until they provide the expected results.

By filtering articles which have a particular structure, it is possible to get statistical data about the repartition of the different types of article structures over the dictionary. Knowing about different structural types, we can further modularize the extraction rules (depending on structure types) and make them more specialized and more selective. In the case of the *OALD*, around 10% of the articles of the electronic edition deviate from any regular structure type which we could determine for this dictionary; this means that one out of ten articles of *OALD3e* would cause trouble in extraction or transformation: for example, nodes are missing or do not have any function, etc.

[8] For attribute names, we basically use those proposed within the ACQUILEX-Project ([Cal et. al 1990]); ACQUILEX has as well analyzed the *OALD*; however, the ACQUILEX anlysis if this dictionary is not as detailed as ours, since the goals of the two projects are different.

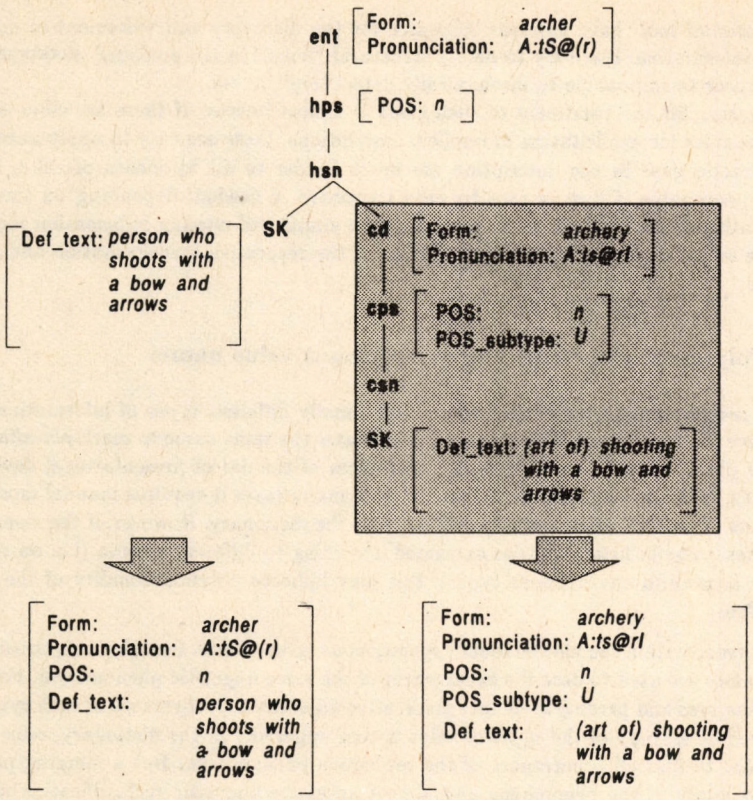


Figure 5: Path information

2.2 Missing or implicit information

The analysis of a sufficient number of articles allows to infer the basic principles of the editorial programme of the dictionary in question, i.e. to know which types of information can be expected in which types of articles (e.g. depending on the category of the lemma).

However, in some articles the expected information is not given explicitly. This may have two reasons: the use of implicit devices or simply an error. A certain type of information may lack consistently and it may then be possible to infer it by rules; in this case, the information is implicit and we have to reconstruct the "rules" or "defaults" which the human user is supposed to apply when reading and "processing" the dictionary article. For example, nominal derivatives with the ending *-tion* do not have a category mark in *OALD*; they are however marked for countability/uncountability. Since such countability marks only apply to nouns, we can reconstruct the category of the *-tion*–nominalizations on this basis.

Quite frequently, information lacks unsystematically. Almost none of the item types is consistently present throughout all relevant articles of *OALD:* This fact has an impact on the interpretation of extraction and transformation results. Here are some examples from *OALD*: 12% of all subentries of the dictionary (compounds, derivatives, idioms) do not have a pronunciation item, and 11% of all verbs do not have an item indicating their syntactic construction potential through a "verb pattern".

Computational tools have different relevance for the discovery and treatment of missing and implicit information: it is easy to detect structural "holes" in the analyzed dictionary articles, but it is near to impossible to mechanically detect implicitness.
The situation for the treatment of such cases is almost inverse: if there are clues allowing to formulate rules for explicitation of implicit descriptions, these are easy to apply automatically. Idiosyncractic gaps in the description are much harder to fill by means of rules, since they are not predictable. Often, a case by case treatment is needed; depending on the relevance and quantity of the material to be amended, the amount of missing information may have an influence on the overall assessment of the use of the respective transformation and extraction exercises.

2.3 Polyfunctional items and synonymous value names

We call *polyfunctional items* those where functionally different types of information indicator items have the same name. For example, *OALD* uses the same name to mark *pluralia tanta* and irregular plural forms. This is typically a problem of the use of presentational devices in the dictionary; it is not very easy to detect, since in many cases it requires manual cross-checking of the lists of partial article texts extracted from the dictionary. However, if the same names of item types occur in lists of entries extracted according to different criteria (i.e. on the basis of different structural environment types), this may indicate polyfunctionality of the respective item names.

The inverse situation can also be found: *synonymous value names.* In such cases, two or more different names are used to describe occurrences of the same linguistic phenomenon. For example, *OALD* has *pred* and *predadj* to denote predicative adjectives. Occurrences of such synonyms are discovered by a study of the types of value names appearing in the dictionary; once identified, it is trivial to find all occurrences of the respective phenomenon. But a merging presupposes extensive study of the phenomena and almost an entire linguistic reclassification of the facts under consideration.

2.4 Statistical significance of items

Many *item values* are very infrequent or even hapaces. Most such cases do not concern particularly idiosyncratic phenomena which could not have been described within any of the "normal" classes of phenomena used anyway in the dictionary; they are, on the contrary, mostly ad hoc descriptions which have not been cross-checked with similar cases during dictionary production. For example, *OALD* has 78 different items indicating the word class. 98% of the occurrences of these are however instances of the usual part of speech types "noun", "verb", "adjective" and "adverb"; the remaining 74 category types make together 2% of the occurrences. This situation is shown in (6), which also illustrates that statistical irrelevance often correlates with synonymous value names.

This type of problems can only be detected with realistic effort if computational tools are used. An inventory of all item values, for example of the word class indicator items, has been set up. All occurrences of the items have been collected and the frequency of the value names across the dictionary has been determined.

selected part of speech values	*count*	*percentage*
n	20985	53.44%
adj	7001	17.83%
vt	4870	12.40%
vi	2788	7.10%
adv	2736	6.96%
adv_of_degree	5	0.01%
adv_of_place_and_direction	1	0.0025%
pron	65	0.16%
pers_pron	6	0.01%
emphat_pron	2	0.0050%
emph_pron	1	0.0025%
reflex_pron	1	0.0025%
interr_pron	3	0.0076%
interr_adv	2	0.0050%
interr_adj	1	0.0025%

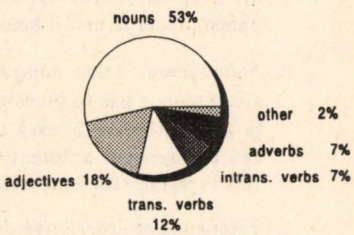


Figure 6: Distribution of *part of speech* attributes.

2.5 Summary: types of problems, computational support for detection and repair

Having discussed some examples of the most frequent problems encountered in the preparation of a transformation of the *OALD* into an attribute-value pair based format, we now try to summarize the types of problems encountered and to assess possibilites of discovering and repairing them by computational means.

The methodological and practical steps of the dictionary analysis start with the article structure. However, in the end, we are interested more in the reuse of the linguistic description conveyed mostly by information indicator items than in an account of the article structure itself is thus a necessary precondition for the more linguistically oriented analysis of the reusability of the descriptive output.

The following typological summary of the most frequent errors encountered is also oriented towards linguistic information types and not towards article structure only. The following types of problems have been predominantly encountered:

1. *Obligatory items lacking* (cf. above, section 2.2): once article structure patterns are determined, it is easy to find out the entries which lack certain items. However, it is almost impossible to automatically repair this problem: this would require new descriptive work. Consequently, a statistical assessment of the frequency of the problem and of the relation between complete and incomplete entries is often decisive for the inclusion of a certain type of item in the set of data supposed to be relevant for reuse.

2. *Implicit indications* (cf. above, section 2.2): the reason for the implicitness of certain indications is the assumed "conviviality" of the dictionary user who can activate dictionary-external knowledge to "process" the entries of the dictionary. A particular type of implicitness is the use of default assumptions in the following sense: if no indication (e.g. of an irregular plural) is given, the applicability of a (sometimes as well implicit) standard pattern (e.g. for plural formation) or of standard properties is assumed. These cases are very hard to detect and to repair by computational means.

3. *Polyfunctional items* (cf. above, section 2.3): this problem comes down to a lack of descriptive differentiation among elements described by one and the same descriptive device. Tools for extraction of substructures and for comparing elements extracted allow to easily get acces to the relevant material. However, reclassification is a descriptive task which needs to be carried out manually. Consequently, an assessment of the frequency of this phenomenon is useful before any transformation.

4. *Synonymous value names* (cf. above, section 2.3): the fact that two value names are synonymous has to be established manually. But a list of all value names used anywhere in the dictionary is easy to establish. Similarly, the comparison of the relevant articles can be prepared automatically, and once it is clear which value names can be "merged", this merging can be carried out by computational means as well.

5. *Frequency of descriptive devices* (cf. above, section 2.4): it is of course easy to extract the relevant types of article structures and of items, and to run statistical tools on the results. Repair mechanisms have to rely on a (manual) reclassification of the data involved. And similarly, it is a matter of linguistic debate to decide whether or not to merge an infrequent type of description with another, more frequent one.

6. *Descriptive errors*: in the reinterpretation process, it may turn out that certain descriptive claims of the dictionary are not accepted by those who wish to reuse the dictionary. Such cases may require reclassification, however, detection and repair are both beyond the scope of computational tool support.

In the following illustration we summarize the abovementioned types of problems and our experience concerning the usability of computational tools for detection and repair:

Type of problem (section)	Computational support for detection	Computational support for repair
obligatory items lacking (2.2)	yes, via structure types	no
implicit indications (2.2)	no	no
polyfunctional items (2.3)	(yes), initial support for manual work	(yes)
synonymous value names (2.3)	no	yes, after extensive preparation
frequency of descriptive devices (2.4)	yes	no
descriptive errors	no	no

3 Conclusion

This contribution deals with methods and tools for a detailed and exhaustive analysis of traditional dictionaries in view of their (re-)usability as information sources for NLP. We are thus interested in the amount and quality of linguistic information extractable from such resources. We discuss a method for the structural analysis of dictionary articles. This method, adapted from WIEGANDS work in metalexicography, serves as a basis for two types of activities; the study of individual types of information in a given dictionary (realized through extraction of certain types of article substructures), and the formulation of transformation rules allowing

to collect information along a given path through an article. Such a method is necessary for the reuse of those dictionaries of which the internal (article structure, textual deployment of articles) is not defined through a formal device, such as a DTD of SGML. This is the case with most dictionaries produced according to the traditional working methodology of lexicography.

We combine the structural analysis with a coherence check on structure types and with the statistical analysis of occurrences of certain types of items. The possibility of collecting evidence for the descriptive devices from all over the dictionary greatly facilitates this task and allows at all for quantifiable statements. The extraction and information routines are implemented as (in part parametric) pattern matching programs in string handling languages. These tools, along with a descriptive linguistic assessment and an exact metalexicographic dictionary description allows to realistically assess the effort needed to re-represent a traditional dictionary for the purpose of NLP.

References

[OALD3] Hornby, A.S. (Ed.): Oxford Advanced Learner's Dictionary, 3rd Edition. Oxford University Press, 1974.

[OALD3e] Hornby, A.S. (Ed.): Oxford Advanced Learner's Dictionary, 3rd Edition, Electronic Version. Oxford University Press, 1988.

[Alshawi et al.1989] Alshawi, Hiyan / Boguraev, Branimir / Carter, David: Placing the dictionary on-line. In: [Bog/Bri 1989].

[Bog/Bri 1989] Boguraev, Bran / Briscoe, Ted (Ed.): Computational Lexicography for Natural Language Processing. London, New York: Longman 1989.

[Cal et. al 1990] Calzolari, Nicoletta / Peters, Carol / Roventini, Adriana: Computational Model of the Dictionary Entry. In: Preliminary Report. Projekt ACQUILEX. Esprit Basic Research Action No. 3030. Pisa, April 1990. ILC-ACQ-1-90. 1990

[Christ 1990] Christ, Oliver. Die Nutzbarmachung eines maschinenlesbaren Standardlexikons durch Transformation in eine CLOS-basierte lexikalische Datenbasis. Studienarbeit Nr. 924 am Institut für Informatik der Universität Stuttgart, 1990.

[Hau/Wie 1989] Hausmann, Franz Josef / Wiegand, Herbert Ernst: Component Parts and Structures of General Monolingual Dictionaries: A Survey. In: [HSK 5.1], pp. 328-360.

[Hei/McN 1991] Heid, Ulrich / McNaught, John. Eurotra–7 Study: Feasibility and Project Definition Study on the Reusability of lexical and terminological resources in computerized applications. Final Report. Under the responsibility of Ulrich Heid and John McNaught. Stuttgart, August 1991.

[Heyn 1992] Heyn, Matthias. Wiederverwendung maschinenlesbarer Wörterbücher. Eine computergestützte metalexikographische Studie zur Wiederverwendung des Oxford Advanced Learner's Dictionary in NLP. erscheint: Tübingen: Niemeyer 1992 (Lexicographica Series Maior).

[HSK 5.1] Wörterbücher, Dictionaries, Dictionnaires. Ein internationales Handbuch zur Lexikographie. Eds: Hausmann, Franz Josef / Reichmann, Oskar / Wiegand, Herbert Ernst / Zgusta, Ladislav. Erster Teilband. Berlin / New York 1989 (Handbücher zur Sprach- und Kommunikationswissenschaft 5.1).

[Wiegand 1989a] Wiegand, Herbert Ernst: Arten von Mikrostrukturen im allgemeinen einsprachigen Wörterbuch. In: [HSK 5.1], pp. 462-501.

[Wiegand 1989b] Wiegand, Herbert Ernst. Der Begriff der Mikrostruktur: Geschichte, Probleme, Perspektiven. In: [HSK 5.1], pp. 409-461.

[Wiegand 1991] Wiegand, Herbert Ernst. Über die Strukturen der Artikeltexte im Frühneuhochdeutschen Wörterbuch. Zugleich ein Versuch zur Weiterentwicklung einer Theorie lexikographischer Texte. In: Ulrich Goebel and Oskar Reichmann (Eds.): Historical Lexicography of the German Language. Volume 2: Lewiston/Queenston/Lampeter: Edwin Mellen 1991 (Studies in German Language and Literature Vol. 6 / Studies in Russian and German Nr. 3), 341-672.

[Wiegand 1990] Wiegand, Herbert Ernst: Printed Dictionaries and Their Parts as Texts. An Overview of More Recent Research as an Introduction. In: Lexicographica 6, 1990. pp. 1 - 126.

Une analyse sémantique et syntaxique des phrases à verbes supports de l'allemand et du français

KATHARINA GREWE

Présentation

Cette étude a pour objet les phrases à verbes supports ou - selon des grammairiens allemands - les 'Funktionsverbgefüge' (FVG)[2] dans la langue française et allemande. Nous nous attacherons à déterminer les différentes structures sémantiques des FVG. Les FVG sont des phénomènes dont les fonctions sont diverses dans la parole. Ils servent à la réalisation des différents aspects ('Aktionsarten') en remplacement de la voix passive et comme un moyen pour réaliser certaines nuances stylistiques. Pour établir le caractère spécifique des FVG, il faut en premier lieu faire une distinction entre les FVG et autres unités phraséologiques. Afin de pouvoir formuler les généralisations adéquates, il est nécessaire d'élaborer des traits grammaticaux qui permettront d'identifier les items lexicaux qui feront l'objet de cet article.

1 Introduction

Cet article se propose d'étudier des constructions nominales prédicatives comme

(a.1) *eine Frage stellen*
(b.1) *in Erwägung ziehen*
(c.1) *in Wut versetzen*

Il est de tradition dans la philologie allemande de désigner ces syntagmes par 'Funktionsverbfügung' (Heringer 1968), 'Funktionsverbgefüge' (Engelen 1968), 'Funktionsverbformeln' (Erben [11]1972), 'Streckformen' (Schmidt 1968), etc. Dans ces phrases, les verbes sont dits 'Streckverben' (Reiners 1943; Häusermann 1977) et 'Funktionsverben' (von Polenz 1963). Même l'élément nominal ne connaît pas un terme unique: 'Gefügenomen' (Engel 1988), 'Funktionsnomen' (Herrlitz 1973), etc.

---

1 Ce travail s'est effectué dans le cadre du projet sur la structure sémantico-syntaxique d'un dictionnaire, de l'Université de Bochum, Sprachwissenschaftliches Institut (Prof. Dr. Helmut Schnelle) et dans le cadre d'une épreuve écrite d'examen à l'Université de Bochum en 1992 sous le titre "Funktionsverbgefüge im Französischen und Deutschen".

2 L'auteur emploie dans cet article les termes allemands de 'Funktionsverb' (FV), de 'Funktionsverbgefüge' (FVG) et de 'Funktionsnomen' (FN).

En français, de telles expressions sont aussi nombreuses:

(a.2) *poser une question*
(b.2) *prendre en considération*
(c.2) *mettre en rage*

Comme le 'discours répété' en général, les FVG du français sont un terrain peu exploré. On manque de terminologie suffisamment large et universellement acceptée. On a généralement considéré les phrases à verbes supports comme un type des unités phraséologiques (Bally 1905). On compte, il est vrai, ces unités complexes appelées 'série verbale', 'périphrase verbale', 'locution verbale', 'tour verbal' et 'phrase à verbe support' parmi le matériel linguistique. Ce qu'on appelle FV en allemand correspond à la notion française de 'verbe pauvre' (Barth 1961), 'verbe signe' (Grevisse [11]1980), 'verbe opérateur' (Giry-Schneider 1978a+b) et 'verbe support' (Daladier 1978). Le phénomène lui-même est très complexe comme les désignations l'indiquent. Quels sont les traits caractéristiques qui distinguent le FVG (a) *mettre en fureur* des syntagmes (b) *mettre à l'ombre* et (c) *mettre (son argent) à la caisse d'épargne* dans les phrases suivantes?

(a) *Luc a mis Jean en fureur*
(b) *Le cambrioleur a été mis à l'ombre*
(c) *Il met son argent à la caisse d'épargne*

L'existence de cette relation en allemand apparaît avec les expressions suivantes du verbe *halten* comme le FVG (a) *ein Versprechen halten* et les constructions (b) *den Mund halten* et (c) *einen Stift halten*. Ces exemples font apparaître que les FVG présentent des structures syntaxiques qui s'appliquent non seulement à des syntagmes libres mais encore à quelques unités phraséo-logiques[3]. Par conséquent ce qui nous importe ici, c'est la question du statut grammatical à attribuer aux FVG. Les FVG prennent pour ainsi dire une position intermédiaire: d'un côté ils ne sont pas des syntagmes libres et d'un autre côté ils ne sont pas non plus des locutions idiomatiques. Les FVG font partie des cas limites de la linguistique.[4]

2 'Funktionsverbgefüge' et unités phraséologiques

Ainsi, la définition traditionelle d'une unité phraséologique qui est toujours la plus répandue, tient compte de l'irrégularité sémantique, de quelques défauts transformationnels, de la stabilité sémantique-syntaxique, de la reproduction, de l'idiomaticité, etc. Ces critères sont partiellement les mêmes pour les FVG. C'est pour cela qu'ils ont seulement de l'importance pour la différence entre des FVG et des mots libres.

Les FVG ne sont pas d'abord produits comme des syntagmes libres dans la parole, mais ils sont classés comme les autres unités phraséologiques du système de la langue. Ils sont entièrement reproduits dans la parole.

Le sens des expressions idiomatiques n'est pas analysable à partir de la signification des éléments constitutifs. La relation entre les signifiés des constituants et la signification de la locution globale est irrégulière. Par exemple *auf die Folter spannen* dans le sens de *jemanden*

---

3 'L'unité phraséologique' doit être comprise comme une notion générale pour tous les types de phraséologismes.

4 "It is possible that we are approaching here the fringe of marginal cases, to be expected in a system as complex as a natural language, where significant systematization is just not possible." (Chomsky 1965:192)

auf etwas gespannt machen, jemandes Neugierde absichtlich nicht befriedigen, le processus d'idiomatisation est total. Aucun des constituants de cette tournure ne conserve sa signification lexicale.

Les phrases figées sont ambigues et l'interprétation littérale (voir les exemples (a.1; a.2)) est toujours possible. Voici quelques exemples:

(a.1) *mettre qqch dans sa poche, montrer les dents à qqn, jeter qqch à la tête de qqn*
(a.2) *etw in die Tasche stecken, jmdm die Zähne zeigen, jmdm etw an den Kopf werfen*
(b.1) *mettre qqn dans sa poche* (familier), *montrer les dents à qqn, jeter qqch à la tête de qqn*
(b.2) *jmdn in die Tasche/Sack stecken* (familier), *jmdm die Zähne zeigen, jmdm etw an den Kopf werfen*

Par contre, le sens d'un FVG résulte de chaque constituant du syntagme. En général, les FVG sont désignés comme unités phraséologiques parce qu'ils sont fixes et forment une unité sémantique. Les lexèmes dans un FVG ne diffèrent pas d'une norme grammaticale lexique. En considération de la subcatégorisation stricte, la plupart des expressions idiomatiques ne sont pas bien formées: *jmdm eins aufs Dach geben, faire semblant.*

Dans les expressions idiomatiques, les substantifs sont normalement définis comme sémantiques 'concrets' (a), ou dans les FVG, ils restent abstraits (b):

(a) *tourner la tête à qqn, marcher sur les talons de qqn, jmdm den Kopf verdrehen, jmdm (dicht) auf den Fersen sein*
(b) *faire un aveu, ein Geständnis ablegen*

3 Critères de classification

Nous allons maintenant ébaucher les propriétés les plus importantes qui ont permis d'isoler les constructions à verbes supports des constructions à verbes pleins et également des constructions à valeur idiomatique.

Les FVG sont des phrases formées d'un FV et un complément d'objet direct ou un complément prépositionnel. Tout d'abord, on va classer les locutions verbales en question selon leur construction superficielle:

- les FVG du type I

  Les FVG du type I (FV+PP) peuvent être classés en trois sous-groupes:

  1. Prép. + Dét. zéro + FN + FV[5]

  (a) *mettre à exécution, entrer en colère, prendre en aversion, être hors de doute, tomber en putréfaction*

  (b) *in Haft nehmen, in Wut bringen, in Betrieb setzen, in Ordnung bringen, in Betracht ziehen, in Erscheinung treten*

  2. Prép. + Dét. indéf. + FN + FV

  (a) *entrer dans une colère terrible, tomber dans une cruelle défaveur, entrer dans une fureur*

---

5 Le schéma tient compte seulement de la composition d'un FVG et pas de la suite des constituants d'une langue spécifique.

(b) *in ein Gespräch eintreten, in eine Panik geraten*

3. Prép. + Dét. déf./Dét. poss. + FN + FV

(a) *mettre dans l'embarras, mettre à l'épreuve, mettre à la raison, mettre à la disposition*

(b) *in den Ruin treiben, in seinen Besitz nehmen, auf der Flucht sein, zur Verarbeitung kommen, zur Durchführung gelangen, zum Lachen bringen*

- les FVG du type II

Les FVG du type II (FV+NP) sont à subdiviser selon la structure superficielle en trois sous-groupes:

1. Dét. zéro + FN + FV

(a) *faire grève, prendre connaissance, rendre hommage, mettre fin, faire attention, avoir de l'admiration*

(b) *Bezug nehmen, Unterstützung geben/finden, Dankbarkeit zeigen, Hilfe leisten, Verwendung finden*

2. Dét. indéf. + FN + FV

(a) *faire un plongeon, faire une promenade, faire un sourire, recevoir une gifle, tirer une conclusion*

(b) *eine Wendung nehmen, eine Untersuchung vornehmen, eine Erklärung abgeben, einen Schrei ausstoßen*

3. Dét. déf./Dét. poss. + FN + FV

(a) *faire l'examen, faire le récit, faire l'achat, prendre la fuite, prendre ses distances*

(b) *der Meinung sein, der Ansicht sein, den Beweis führen, die Flucht ergreifen, den Schluß ziehen*

Mais cette représentation syntaxico-morphologique des FVG ne contribue pas beaucoup à différencier les FVG des locutions idiomatiques ainsi que des unités syntaxiques dites libres.

Le FV n'est qu'un outil morphologique et syntaxique qui sert pratiquement à conjuguer le FN. Ces verbes sont des marques de temps, de personne et de nombre. Il n'y a pas de verbes quelconques qui peuvent avoir la fonction d'un FV dans un FVG. Souvent il s'agit de verbes polysèmes, c'est-à-dire de verbes avec plus de significations. En français, les verbes *donner, pousser, faire, prendre, rendre, avoir, être, aller, mettre, entrer, tomber, tirer* et *demander* sont les verbes les plus utilisés de la catégorie FV. En allemand, les verbes *bringen, kommen, sein, stehen, geraten, setzen, stellen, halten, nehmen, bleiben* et *haben* sont les FV les plus utilisés.

Avec ces verbes, il y a des substantifs qui, malgré leur place et leur forme apparente, ne sont pas des arguments du verbe principal. Du point de vue sémantique, un nom (FN) en relation avec un FV est porteur du sens lexical de l'expression. Le prédicat représenté par le nom prédicatif désigne des actions, des activités, des événements, des états, etc. Les FN sont des substantifs abstraits, dérivés d'un verbe ou d'un adjectif. En illustration les exemples suivants:

(a.1) *prendre la fuite*
(a.2) *entrer en balance*
(b.1) *être en colère*
(a.3) *zum Verschwinden bringen*

(a.4) *in Schwung bringen*
(b.2) *in Verlegenheit bringen*

Comme le montrent les exemples ci-dessus, les FN de ces FVG décrivent presque toujours des actions et des situations ((a.1)-(a.4)). Seulement deux substantifs abstraits démontrent un état ou une propriété ((b.1); (b.2)).

En comparaison à son verbe de base ou à son adjectif de base, le FVG se différencie souvent, soit dans son aspect, soit dans la signification du verbe principal:

- les FVG 'sémantiques' sont différents des lexèmes principaux

  (a) *mettre en danger (être dangereux), faire un récit (réciter)*
  (b) *in Verlegenheit bringen (verlegen sein), zu Ansehen bringen (ansehen)*

- les FVG 'simples' comme synonymes des lexèmes principaux

  (a) *mettre en doute (douter de), donner une réponse (répondre à)*
  (b) *in Zweifel ziehen ((be)zweifeln), Antwort geben (antworten)*

4 Données de combinatoire lexicale

Une particularité des FVG est sa tendance de former une autre expression avec un FV comme élement stable. Le FV peut se construire avec d'autres FN de séries d'oppositions plus ou moins complètes.

prendre	*treffen*
prendre une décision	*eine Entscheidung treffen*
prendre un arrangement	*eine Vereinbarung treffen*
prendre des dispositions	*Vorkehrungen treffen*

prendre	*ergreifen*
prendre la fuite	*die Flucht ergreifen*
prendre l'initiative de	*die Initiative ergreifen*
prendre des mesures	*Maßnahmen ergreifen/treffen*
prendre possession de	*Besitz ergreifen von*
prendre des précautions	*Vorsichtsmaßnahmen ergreifen/treffen*
prendre la parole	*das Wort ergreifen*

Mais il y a des FVG qui peuvent être construits par le FV comme élément remplaçable et par le FN comme élément stable:

(a.1) *être, mettre, entrer en action*
(a.2) *être, mettre, maintenir en marche*
(b.1) *in Bewegung sein, bringen, kommen*
(b.2) *in Gang sein, setzen, halten*

Les FV et les FN également ont des extensions lexicales. On ne discutera pas ici de la question beaucoup trop générale de la paraphrase et de la synonymie. Les significations ne diffèrent pas beaucoup:

(a.1) *donner/intimer l'ordre*
(a.2) *donner/fournir une réponse*
(b.1) *Befehl geben/erteilen*
(b.2) *eine Antwort geben/erteilen*

Les FVG ne sont pas une catégorie homogène. Ils se différencient en divers degrés de la stabilité de leurs composants. Les FVG sont plus ou moins des expressions figées, c'est-à-dire qu'il existe des expressions plus lexicalisées (a) et aussi des expressions moins stables (b):

(a.1) *entrer en ligne de compte, tenir conseil*
(b.1) *mettre à la disposition, tomber dans la misère*
(a.2) *in Frage kommen, zu Rate gehen*
(b.2) *zur Verfügung stellen, in Not geraten*

5 Données sémantiques

5.1 'Funktionsverb' et verbe simple

En fin de compte, ce serait sans doute l'étude sémantique qui devrait fournir le critère décisif. Il y a en effet un ensemble de propriétés sémantiques qui différencient les FVG.

Chaque FVG forme une unité sémantique indépendante qui sert de verbe dans la phrase. Certains noms forment avec des verbes tels que *faire*, *avoir*, etc. des locutions verbales équivalentes à des verbes simples. La substitution d'un FVG à un verbe de même radical prouve sa cohésion sémantique. Ainsi, le FVG est utilisé comme synonyme du verbe de base. Considérons la périphrase verbale *Max fait des compliments à Léa*; elle est synonyme de la phrase à verbe *Max complimente Léa.*

(a.1) *prendre en considération* et *considérer*
(b.1) *in Erwägung ziehen* et *erwägen*
(a.2) *poser une question* et *questionner*
(b.2) *eine Frage stellen* et *fragen*

- les FVG sont remplaçables par le verbe principal sans éprouver une perte d'information:

  (a.1) *porter un jugement sur qqch/qqn (juger)*
  porter une (grande) admiration (admirer)
  (a.2) *prendre en haine (hair)*
  (a.3) *Nachricht geben (benachrichtigen)*

- les FVG sont normalement les seules possibilités d'exprimer l'action verbale:

  1. les FVG n'ont pas d'équivalents synthétiques

     (a.1) *mettre au courant (informer)*
     (a.2) *mettre en danger (compromettre)*
     (b.1) *in Kenntnis setzen (mitteilen)*

  2.

  a) le verbe de base est désuet ou inutilisé

     entrer en colère (vieilli: se colérer)

  b) le FVG est désuet ou inutilisé

     (vieilli) mettre en oubli (oublier)
     (veraltet) in Harnisch bringen (wütend machen)

3. les FVG expriment des significations essentielles contrairement à leur verbe principal:

a) le FVG a une signification spéciale

(a) *mettre en observation (observer)*
(b) *unter Beobachtung stellen (beobachten)*

b) le FVG a une signification générale

entrer dans le sommeil (sommeiller)

Une indication poussée de l'unité sémantique d'un FVG est la nominalisation de l'expression. Mais cette opération n'est pas toujours usuelle:

(a.1) *la mise en service*
(b.1) *die Inbetriebnahme*
(a.2) *la mise en état*
(b.2) *die Instandsetzung*

Beaucoup de FV (a.1; b.1) signalent une signification à voix passive (a.2; b.2):

(a.1) *qqch entre en accomplissement*
(a.2) *qqch est accompli*
(b.1) *etw kommt zur Vollendung*
(b.2) *etw wird vollendet*

5.2 Phases du procès et causativité

La principale fonction sémantique réside dans la précision ou la modification d'une information sémantique spécifique exprimée par le FV comme les FVG *in Gang sein/être en marche, in Gang kommen/entrer en marche* et *in Gang bringen/mettre en marche* le montrent. Ces modifications sont nommées en philologie par le terme ´Aktionsart´ et elles se manifestent en signes comme durée, finalité, causativité et leurs combinaisons. La notion allemande d'´Aktionsart´ est difficile à traduire en français. Elle a été confondue avec celle d'aspect. Pour éviter les confusions des termes d'´Aktionsart´ et d'aspect, nous avons préferé la notion allemande dans cet article.

Les FVG se répartissent en trois sous-groupes, selon qu'ils expriment (a) l'entrée dans un état ou le début d'une action, (b) l'action en déroulement et (c) le changement d'état provoqué. Le verbe principal (d) est déterminé par une réalisation certaine d'un procès.

(a) *entrer en marche, in Bewegung kommen*
(b) *être en marche, in Bewegung sein*
(c) *mettre en marche, in Bewegung setzen*
(d) *marcher, (sich) bewegen*

Dans l'échange d'un FV avec une variante opposée, la signification varie:

(a.1) *avoir de l'aversion pour qqn, avoir un certain balancement*
(a.2) *prendre en aversion, prendre un certain balancement*
(a.3) *garder un certain balancement*
(a.4) *perdre (tout + son) balancement*
(b.1) *avoir peur*
(b.2) *prendre peur*
(b.3) *faire peur*
(c.1) *in Bewegung sein, in Bewegung bleiben*

(c.2) *in Bewegung setzen, in Bewegung bringen, in Bewegung versetzen*
(c.3) *in Bewegung kommen, in Bewegung geraten*

Elles expriment alors les différentes phases d'un procès:

- la durativité
- la transformativité

La durativité démontre le déroulement d'un procès acutel. La durativité est représentée par des verbes tels que *être, avoir, rester, demeurer, haben, sein* et *bleiben.*

(a) *rester en ordre, être en marche, être en colère, être à la disposition, avoir à sa disposition*
(b) *in Ordnung bleiben, in Gang sein, in Wut sein, zur Verfügung stehen, zur Verfügung haben*

Le mode d'action du type transformatif exprime le pasage d'un procès à un autre ou limite le déroulement du procès. La transformativité se divise en deux *Aktionsarten*:

- l'inchoativité[6]

Un verbe est alors qualifié d'inchoatif si son début est marqué (entrée en l'état). La phase inchoative est identifiée par les verbes suivants: *tomber, entrer, kommen* et *geraten.*

(a) *prendre peur, entrer en relation, prendre connaissance, entrer en balance, entrer dans une colère noire*
(b) *in Angst geraten, in Verbindung treten, Kenntnis nehmen, in blinde Wut geraten*

- l'égressivité[7]

Le mode d'action du type égressif marque la fin de l'action ou le résultat d'un procès:

(a) *eine Verbesserung herbeiführen, eine Untersuchung durchführen*
(b) *apporter une amélioration,* réaliser *une étude*

Les FVG servent aussi à exprimer un fait de causativité. La causativité, contrairement à ce qui est écrit dans de nombreux travaux sur les ´Aktionsarten´, n'est pas une ´Aktionsart´ si l'on entend par ce terme le mode temporel de déroulement du procès.

(a) *mettre à la disposition, mettre en mouvement*
(b) *zur Verfügung stellen, in Bewegung bringen*

Elle correspond le plus souvent à l'introduction d'un argument supplémentaire de la phrase simple de départ (von Polenz 1987; Persson 1975):

(a.1) *Le moteur marche*
(a.2) *Il met le moteur en marche*
(b.1) *Der Motor läuft*
(b.2) *Er bringt den Motor in Gang*

---

6 Appelé aussi "ingressif".

7 Appelé aussi "résultatif".

La causativité peut être assimilée à un trait facultatif pouvant se combiner avec chacune des ´Aktionsarten´. Par exemple les verbes causatifs *mettre* et *bringen* du type I sont très productifs en français et en allemand:

(a.1) *mettre en colère*
(a.2) *mettre en sûreté*
(b.1) *in Wut bringen*
(b.2) *in Sicherheit bringen*

L'application de l'opérateur causatif ne s'appliquent qu'aux certains verbes. Giry-Schneider (1986:55-56) a ainsi montré qu'il y a une relation entre les phrases à *avoir* et *il y a* ou à *faire* et *donner*:

(a.1) *Max a faim*
(a.2) *Ceci donne faim à Max*
(b.1) *Paul a eu un choc*
(b.2) *Cette nouvelle a fait un choc à Paul*
(c.1) *Max hat Hunger*
(c.2) *Das macht ihm Hunger -> Das macht ihn hungrig*
(d.1) *Paul hat einen Schock (gehabt)*
(d.2) *Diese Nachricht hat bei Paul einen Schock verursacht/ausgelöst*

Bibliographie

Bally, Ch. (1905): Précis de stylistique. Esquisse d'une méthode fondée sur l'étude du français moderne. Genève: A. Eggimann.

Barth, G. (1961): Recherches sur la fréquence et la valeur des parties du discours en français, en anglais et en espagnol. Paris: Didier.

Chomsky, N. (1965): Aspects of the Theory of Syntax. Cambridge/Mass.: The MIT-Press.

Daladier, A. (1978): Problèmes d´analyse d´un type de nominalisation en français et de certains groupes nominaux complexes. Thèse de doctorat de 3e cycle. L.A.D.L. Paris.

Engel, U. (1988): Deutsche Grammatik. Heidelberg: Groos.

Engelen, B. (1968): "Zum System der Funktionsverbgefüge." Wirkendes Wort 18, 289-303.

Erben, J. ([11]1972): Deutsche Grammatik. Ein Abriß. München: Hueber.

Giry-Schneider, J. (1978a): Les nominalisations en français. L´opérateur "faire" dans le lexique. Genève: Droz (Langue et Cultures 9).

Giry-Schneider, J. (1978b): "Interprétation aspectuelle des construnctions verbales à double analyse." Linguisticae Investigationes II, 1, 23-53.

Giry-Schneider, J. (1986): "Les noms construits avec *faire*: compléments ou prédicats?" Langue Française 69, 1, 49-63.

Grevisse, M. ([11]1980): Le bon usage. Grammaire française avec des remarques sur la langue française d´aujourd´hui. Paris; Gembloux: Duculot.

Grewe, K. (1992): Funktionsverbgefüge im Französischen und Deutschen. Magisterarbeit am Institut für Romanische Philologie. Ruhr-Universität Bochum.

Gross, G. & Vivès, R., eds. (1986): "Les constructions nominales et l'élaboration d'un lexique-grammaire." Langue Française 69, 1, 5-27.

Gross, M. (1981): "Les bases empiriques de la notion de prédicat sémantique." Langages 63, 7-52.

Häusermann, J. (1977): Phraseologie. Hauptprobleme der deutschen Phraseologie auf der Basis sowjetischer Forschungsergebnisse. Tübingen: Niemeyer (Linguistische Arbeiten 47).

Heringer, H.-J. (1968): Die Opposition von "kommen" und "bringen" als Funktionsverben. Untersuchungen zur grammatischen Wertigkeit und Aktionsart. Düsseldorf: Pädagogischer Verlag Schwann (Sprache der Gegenwart 3).

Herrlitz, W. (1973): Funktionsverbgefüge vom Typ "in Erfahrung bringen". Ein Beitrag zur generativ-transformationellen Grammatik des Deutschen. Tübingen: Niemeyer (Linguistische Arbeiten 1).

De Negroni-Peyre, D. (1978): "Nominalisation par "être en" et réflexivation." Lingvisticae Investigationes II, 1, 127-164.

Persson, I. (1975): Das System der kausativen Funktionsverbgefüge. Eine semantisch-syntaktische Analyse einiger verwandter Konstruktionen. Lund: CWK Gleerup (Lunder germanistische Forschungen 42).

Polenz, Peter von (1963): Funktionsverben im heutigen Deutsch. Sprache in der rationalisierten Welt. Düsseldorf (Beiheft zur Zeitschrift Wirkendes Wort 5).

Polenz, Peter von (1987): "Funktionsverben, Funktionsverbgefüge und Verwandtes. Vorschläge zur satzsemantischen Lexikographie." Zeitschrift für Germanistische Linguistik 15, 169-189.

Reiners, L. (1943): Stilkunst. Ein Lehrbuch deutscher Prosa. München: Becksche Verlagsbuchhandlung.

Schmidt, V. (1968): Die Streckformen des deutschen Verbums. Substantivisch-verbale Wortverbindungen in publizistischen Texten der Jahre 1948-1967. Halle (Saale): VEB/ Niemeyer.

Budapest Sociolinguistic Interview — A Corpus of Spoken Hungarian

ILONA KASSAI

On the project

In 1985 a large-scale sociolinguistic research project, called the Survey of Spoken Hungarian (SSH) was decided and organized by the Linguistics Institute of the Hungarian Academy of Sciences.
The project aims

- to complement and modify existing descriptions of Hungarian which are based mostly on written sources or the intuitions of individual linguists through an analysis of a massive corpus of spoken material as well as elicitation experiments;

- to analyse the linguistic variations to be found in the speech of speakers in Budapest belonging to various socio-economic groups with a precise sociological profile;

- to examine various language styles, that is variations subject to the amount of speakers' audio-monitoring of their speech;

- to accumulate data that would allow for the analysis of linguistic change if similar data are collected in 10-20-30 years' time;

To achieve the objectives outlined above, various research tools are needed, the sociolinguistic interview being one of them. It undertakes to provide solid empirical evidence to answer selected questions concerning phonological, morphological, syntactical and lexical variables. Obviously, the spoken corpus makes it possible to investigate several problems not directly targeted in the project or not even thought of today.

On the selection of research questions

- In summer 1986 our fellow researchers at the Linguistics Institute of the Hungarian Academy were asked to state what they considered important questions for our project to focus on. The answers were all processed.

- In a series of project meetings in 1986, our group discussed work in related projects abroad -- relevant conclusions in the literature were discussed and adopted in our work.

- We asked for recommendations on data collection in the field of phonology and syntax.

- On the basis of the above information Miklós Kontra, the leader of the project defined the list of research questions.

- The list so derived was pruned to include only the phenomena amenable to a sociolinguistic interview.

On the selection of informants

In Autumn 1987 50 pilot interviews were conducted with a quota sample of 10 teachers over 50 years of age, 10 university students, 10 shop assistants, 10 blue-collar workers and 10 vocational trainees aged 15-16. This phase of our project is called the Version Two of the Budapest Sociolinguistic Interview.

In 1988-89, 200 taperecorded interviews were completed in Budapest. This phase of our study is called the Budapest Sociolinguistic Interview Version Three. The 200 informants we interviewed form the Budapest subsample of a random stratified national sample that is representative of the entire country by age, sex, schooling and type of settlement. The 850 people who responded to a broader opinion survey entitled "Social stratification - communicative stratification" (cf. Angelusz & Tardos 1987) also answered questionnaires about language usage (for more details about the national survey see Kontra 1992). Therefore, it can be claimed that the Budapest sample meets any sociological standard.

It is a special chance that due to drastic social and economic changes in Hungary we gathered data about both the final state of Hungary under communism and the baseline against which future developments of the language can be measured.

The structure of the Budapest Sociolinguistic Interview

The Budapest Sociolinguistic Interview itself was constructed along the lines defined by Labov (1966, 1972, 1984) and includes forms of reading, minimal pairs, elicitation procedures and, as a result of a network of conversational modules, relatively open-ended conversation. The average duration of an interview is 2.5 hours, of which at least 30 minutes is guided conversation. Owing to the intricate nature of the testing involved, not only is it the case that a single test sentence may examine a number of different research questions but also the same research question may be involved in a number of different test sentences (as well as in the guided conversations, of course).

Processing of data

The project takes advantage of computer technology for storing, coding, and analyzing the transcribed texts.
Computationally, the data collected through the interview fall into two broad categories (1) test-like tasks and (2) continuous speech. The two types of data complement each other: without the test results we could not make comparative analysis across informants, whereas without continuous speech data we could not analyse such characteristics of particular elements in speech as their frequency, contextual dependency etc..

Targeted lexical issues

In the rest of the paper I shall briefly present the targeted lexical questions of the Budapest Sociolinguistic Interview.

1. Pronunciation variants
Three questions are asked:
- What is the social distribution of certain stigmatized pronunciation varieties e.g. [inektsio:] 'injection', [sofia:ne:] 'Sopianae'?

- What pronunciation variants do old loans like *nylon* and some recent ones e.g. *spray*, *juice* have ?

- What is the pronunciation of words which show variation in vowel length in speech but are consistently spelt with a long vowel such as *színház* 'theatre', *útiköltség* 'travel costs', *háború* 'war', *fésű* 'comb' and *hűvös* 'cold' as well as *bölcsőde* 'crèche'?

In order to establish the linguistic insecurity index of the informants with regard to pairs of variants, two listening tests administered by means of a Walkman cassette player and headphones are used: the first asks for the correct form (Which is

correct?), while the other investigates the forms used by the informant (How do YOU usually say it?). The discrepancies show the linguistic insecurity of the informant.

2. *Felolt* and *felgyújt*
The verb *felolt* 'extinguish up' used in the sense of 'turn on /the light/' is a semantic anomaly, but frequently used. *Felgyújt* is the correct standard form. The frequency of the deviant usage is investigated by means of the so-called reporter's test (cf. Ball 1986) consisting in that the informant does a running commentary of what the field worker acts out.

3. Meaning interpretation
What does the word *demográfia* 'demography' mean?
Demográfiát akarunk? Meg kell adóztatni a gyerekteleneket 'Do we want demography? Childless couples should be taxed then.' -- a speaker on a live TV program said on 10 December 1986. The speech of more or less uneducated speakers often shows the use of certain fashionable words with transferred meaning, e.g. Nincs egyetértés a politika és az írók között, 'there is no agreement between the policy and writers' i.e. between politicians and writers. It may be presumed that the spread of such "inaccuracies" correlates with the educational background and/or socio-economic status of speakers in that the more educated speakers interpret the word in its literal sense whereas less educated speakers allow transferred senses as well. Research tool: the following multiple choice card.

What does *demográfia* mean?
1) A policy which should result in increasing numbers of births.
2) The study of the changes in numbers of births, deaths, marriages and diseases in a community over a period of time.
3) Both meaning 1) and meaning 2) are correct, this word has two meanings.

5. Meaning definition
As a consequence of the social and economic changes having been carried out in the East European area many foreign words appeared in the mass media. The interview undertakes to explore the informants' knowledge of the meaning of three such words, namely *pluralizmus* 'pluralism', *glásznoszty* 'glasnost' and *peresztrojka* 'perestroika'. The way the task is performed is to get informants give explicit meaning definitions.

4. *Tűzőkapocskiszedő*. The Staple remover test.
The Hungarian equivalent of stapler is *'fűzőgép'* or *'tűzőgép'*. The Comprehensive English-Hungarian Dictionary by Országh lists staples as *'fűző(gép)kapocs'*, *'papírfűző/ könyvfűző drótkapocs'* or simply *'kapocs'*. Staple removers, the handy devices that serve to remove the staples easily and without damaging the hand or the paper, were unknown in 1986 in Hungary. They made their first appearance in stationery stores in early 1987. In June 1987 shop assistants in a stationery store in

the business center of Budapest put down the name of the article on the receipt as *tűzőgép* or *tűzőkapocskiszedő*. This term was used only on the receipt as the goods themselves were sold unpacked, without any brand name and description of the article. In short: we are currently witnessing the spread of a new device at a time when, apart from the official register of goods, the object practically is without a name. Therefore, we have a unique opportunity here to catch the birth of a word in statu nascendi.

Research procedure:

1. The field worker shows staple remover to informant and asks "Have you ever seen such a thing?"
2. Field worker to informant: "What is this?"
3. If answer is "I don't know what it is", then field worker says: "OK, I'll show you what it's for". Then he demonstrates how the remover is used.
4. Field worker: "Now, what is this?"
5. Field worker holds up stapler and asks: "What is this?"
6. Field worker holds up staple and asks: "What is this?"
7. Finally field worker holds up staple remover and asks for a name again.

6. Dialectal words

Within the conversational modules of the interview questions related to language e.g. where is the best Hungarian spoken?, which part of society speaks the best Hungarian?, is a dialect speaker hindered by his speech in upward social mobility? are obligatorily asked. Among the informants in-migrants are addressed an extra question whether they can recall words they had acquired in their native surroundings and which are unknown to the inhabitants of Budapest. The aim is to explore the mental lexicon of in-migrant informants.

References

Angelusz, Róbert & Tardos, Róbert. 1987. Kulturális-kommunikációs rétegződés (Kutatási tervezet). Szociológia, 1987/2:209-231.

Ball, Martin J. 1986. The reporter's test as a sociolinguistic tool. Language in Society 15:375-386.

Kontra, Miklós. 1992. Sociolinguistics in Budapest. in Kontra, Miklós and Tamás Váradi eds. Studies in Spoken Languages: English, German, Finno-Ugric, 83-94. Budapest: Linguistics Institute, Hungarian Academy of Sciences.

Labov, William. 1966. The Social Stratification of English in New York City. Washington, D.C.: Center for Applied Linguistics.

--- 1972. Sociolinguistic Patterns. Philadelphia. University of Pennsylvania Press.

--- 1984. Field Methods of the Project on Linguistic Change and Variation. in Baugh, John and Joel Sherzer eds. Language in Use: Readings in Sociolinguistics, 28-53. Englewood Cliffs, N.J.: Prentice-Hall.

Computational work on the Student's Illustrated Dictionary of Hungarian and the Computational study of its vocabulary

GÁBOR G. KISS

The Publishing House and Printing Press of the Hungarian Academy of Sciences is going to publish the Students' Illustrated Dictionary of Hungarian (henceforth: SIDict) before Christmas 1992. This dictionary for pupils (age 10-16) is being written in the Research Institute for Linguistics by a team of 7 scholars. According to the plans, the dictionary is going to contain 14.000 vocabulary entries as well as 80 pages of colored illustrations (BÍRÓ, 1991).

When comparing it with dictionaries of similar character for other languages, we can say that as regards its vocabulary and complexity SIDict has an intermediate position between the *OXFORD Basic English Dictionary* (10.000 entries) and the *LAROUSSE débutants* (20.000 entries) (OXFORD, 1981; LAROUSSE, 1986).

Parallel with the publication of this volume by Christmas 1992, a floppy-disc version for IBM PC/AT will made available for pupils, lexicographers as well as experts interested in further developments. The size of the dictionary will be 2.8 Mbyte.

The Publishing House and Printing Press of the Hungarian Academy of Sciences has arranged, in order to speed up the work, that the writing and the preparation of lexical entries for printing should go simultaneously, with a minimal delay. This was achieved by using computers. This way the Publishing House has achieved that not much after the preparation of the last dictionary entry the camera-ready set material of the SIDict will available.

When preparing the SIDict, IBM PC/ATs and XTs working under DOS help the writers of the dictionary in the following ways:

1. Compiling the dictionary entry list:

We used computers when compiling the dictionary entry list. We typed an amplified dictionary entry list into the computer. The amplified entry list based on a preliminary selection contains 16.000 lexical entries. * denotes compound lexical entries, e.g. *nyak*leves, idő*álló*. The dictionary entry list was created by WordPerfect. Using the SORT command, we ordered the dictionary entry list in an alphabetical order according to the last element of compounds. Thus we got lists like

```
1) esküdt*szék,     7)      tan*szék,      1)  film*szerű,
2)  forgó*szék,     8)  törvény*szék,      2) közép*szerű,
3)  hinta*szék,     9)      úri*szék,      3) sport*szerű,
4) iskola*szék,    10) villamos*szék,      4)  szak*szerű,
5)  pohár*szék,    11)  zongora*szék,
6)    szó*szék,
```

This list was used when preparing the last version of the dictionary entry list, i.e. when reducing the preliminary list containing 16.000 entries to the final version of 14.000 entries.

2. Typographical preparation of SIDict by the help of computers:

The Publishing House and Printing Press of the Hungarian Academy of Sciences works on FERRANTI printing equipment and composing system. In the previous years the possibility of a novel input to these typographical equipment was designed and elaborated. Through this input coded texts containing ASCII characters can be entered in the composing system. The codes refer on the one hand to the letter types, and on the other to the special characters used in the dictionary. Codes are given, as it can be seen from a sample below, in brackets < >:

```
normal letter type  = <1> [ xxxxxxxx ]
italics letter type = <2> [ xxxxxxxx ]
bold letter type    = <3> [ xxxxxxxx ]

tilde clinging to a suffix [ ~xxx ] = <50>
tilde with a comma         [ ‘xxx ] = <51>
tilde standing alone        [ _~_ ] = <52>

... = \q\q\q
* = <66>
| = <6>
```

The dictionary entries of SIDict were stored in the computer using these codes. An example for the dictionary stored for the printing press:

```
<5><41>híre-hamva<40> szragos fn (csak esz 3. személyben,
ált. alanyesetben)
<4><2>A tegnap leesett hónak ma már <52> sincs, <1>nyoma
sincs, eltűnt.

<5><41>híres<40>
<4><3>I. <1>mn <2><50>ek, <50>t <1>v. <2><50>et, <50>en
<3>1. <2><52> ember, író, politikus, színész:
<1>kiemelkedő tulajdonságairól, teljesítményéről jól
ismert (= hírneves). <2><52> regény, dal, város:
<1>közismert, nevezetes. (biz) <2>Nem valami <52> ez a
dolgozat, <1>gyenge minőségű, középszerű. <3>2.
<1>(pejor) <2>Megint itt vannak a <50>, <1>rossz hírű (=
hírhedt) <2>barátaid.
<3>II. <1>fn <2><50>ek, <50>et, <50>e <1>(pejor)
```

```
<1>Kétes hírű személy. <2>Hol az a <50>?
<1>(Megszólításban:) <2>No, te <50>!

<5><41>híresség<40> fn <2><50>ek, <50>et, <50>e
<4><3>1. <2>Apjának <50>e, <1>híres volta <2>nemcsak
segítette, akadályozta is pályáján. <3>2. <2>Az előadásra
meghívták az összes <50>et, <1>híres embert.
```

The SIDict was stored on the computer in WordPerfect files. In this work we utilized the vast macro-programming possibilities of the WordPerfect program to a great extent (KISS, 1991). We assigned the most common codes consisting of several characters to keyboard macros. This had a twofold advantage: on the one hand, typing and storing was quicker and, on the other, the character sequences of the codes appeared without any misspellings in the texts. E.g.

```
Alt key + o letter = (pej) [=(ironic)]
Alt key + b letter = <1>(=
Alt key + g letter = <1>v. <2> [<1> or <2>]
Alt key + a letter = <40> ige <2> [=\<40> verb <2>]
```

The writers of the dictionary asked us to enable them to check the texts, which had been stored in the computer's memory, in a non-coded form. The authors wanted to see and correct the text in the same form as it will be printed out in its final version. We were not able to have all parts of the dictionary printed out several times for proof-reading (this was due, among other factors, to constraints of time and budget). Thus before having the text printed for the authors, we used a series of WordPerfect macros entitled PREPA.WPM to convert the text full of codes to the common typographical form. In the following we exemplify this by showing the program listing of the macro that converts entries starting with <2> to italics.

```
Macro:       Action
File:        C:\WP51\MACROS\PREPA2.WPM
Description: A <2> kód utáni szöveg italizálása
```

```
{DISPLAY OFF}
{Home}{Home}{Up}
{LABEL}kezdet~
{Search}<2>{Search}
{Block}
{Search}<{Search}{left}
{Font}22
{ON NOT FOUND}{GO}vege~~
{GO}kezdet~
{LABEL}vege~
{Home}{Home}{Up}
```

By using the macro-program series we got the following printed version for the coded text seen above:

híre—hamva szragos fn (csak esz 3. személyben, ált. alanyesetben)
A tegnap leesett hónak ma már ~ sincs, nyoma sincs, eltűnt.

híres
I. mn *~ek, ~t* v. *~et, ~en*
1. *~ ember, író, politikus, színész:* kiemelkedő tulajdonságairól, teljesítményéről jól ismert (= hírneves). *~ regény, dal, város:* közismert, nevezetes. (biz) *Nem valami ~ ez a dolgozat*, gyenge minőségű, középszerű. **2.** (pejor) *Megint itt vannak a ~*, rossz hírű (= hírhedt) *barátaid.*
II. fn *~ek, ~et, ~e* (pejor)
Kétes hírű személy. *Hol az a ~?* (Megszólításban:) *No, te ~!*

híresség fn *~ek, ~et, ~e*
1. *Apjának ~e*, híres volta *nemcsak segítette, akadályozta is pályáján.* **2.** *Az előadásra meghívták az összes ~et*, híres embert.

These is the final form of SIDict from the Publishing House and Printing Press of the Hungarian Academy of Sciences before last correction:

láb fn *~ak, ~at, ~a*
1. Ember, szárazföldi állat járásra való végtagja, ill. ennek alsó része; lábfej. *Töri a cipő a ~át. ~a kel:* eltűnik, nyoma vész; *nagy ~on él:* pazarló, költekező életmódot folytat. **2.** Tárgynak az(ok) az oszlopszerű része(i), amely(ek)en áll. *Az asztal, a zongora ~a.* **3.** Hegy(ség), halom stb. alsó része. *A hegy ~ánál áll a ház.* **4.** Régi, ill. külföldi hosszmérték: kb. 30 cm. *Hat ~ magas.*
lábad ige *~ni* (vál)
Könnybe ~ a szeme, könnyes lesz.
lábadoz|ik ige *~ni*
Súlyos betegségből gyógyulóban van.
lábal ige *~ni* **lábol**
Sekély vízben, sárban, hóban – lábát nehezen emelgetve – jár, megy. *A pocsolyában ~. A pocsolyát ~ja.* (= gázol)
labanc fn *~ok, ~ot, ~a* (tört)
1. A kurucok ellen harcoló császári zsoldos. **2.** (pejor) Habsburg-párti magyar.
lábas[1] mn *~ak, ~at, ~a*
Lábon, talapzaton álló. *~ óra. ~ ház:* árkádos.
lábas[2] fn *~ok, ~t, ~a* **lábos**
Alacsony, kétfülű főzőedény.
lábatlankod|ik ige *~ni* (biz)
Másnak útjában van, ténfergésével zavarja, láb alatt van.
lábazat fn *~ok, ~ot, ~a*
Bútornak, építménynek, falnak, szobornak stb. a legalsó, sajátosan kiképzett része.
lábbeli fn *~k, ~t, ~je*
Az öltözéknek a lábon viselt tartozéka: cipő, csizma, szandál, papucs stb.
ra. **2.** Fölfelé nyíló fedelű, alacsony bútor ruhanemű, élelmiszer stb. tárolására. *Tulipán(t)os ~:* régi parasztházak festéssel (ritkábban faragással) díszített jellegzetes bútordarabja. **3.** Nagyobb doboz.
ladik fn *~ok, ~ot, ~ja*
Lapos fenekű csónak. *„Általmennék én a Tiszán ladikon, Ladikon, de ladikon"* (népdal).
lagúna fn *~k, ~t, ~ja*
Homok- v. korallszigetekkel (részben) elzárt sekély tengerrész. *A ~k városa:* Velence.
lagzi fn *~k, ~t, ~ja* (nép, biz)
Lakodalom.
lágy
I. mn *~ak, ~at, ~an*
1. Könnyen formálható, laza szerkezetű, sokszor nedvességet tartalmazó (= puha, ≠ kemény). *~ tojás:* héjában hígra főzött. *Az ólom ~ fém. A ~ szárú növénynek nincs fás része.* **2.** *~ víz:* kevés ásványi sót tartalmazó. **3.** Nem határozott, nem éles, nem erős, nem éles körvonalú. *~ vonások, ~ hang, ~ fém, ~ hullámok, ~ szellő.* **4.** Szelíd, gyöngéd, engedékeny. *~ szíve van, ~an cirógat.*
II. fn *~ak, ~at, ~a*
Vminek a ~a: a lágyabb része. *A feje ~a:* a koponyacsontok találkozásának kisgyermekkorban porcos része. *66 Benőtt a feje ~a:* megkomolyodott.
lágyék fn *~ok, ~ot, ~a*
A hasfalnak alulról a csípőkig terjedő, háromszög alakú része.

3. Editing SIDict by using DATABASE I:

The finished dictionary entries of SIDict were placed continuously into a textual data base, which will be referred to DATABASE I. The GREP program was used to look up any character sequences in their contexts which the editors and/or authors of SIDict requested. This way this we gave an effective help both to the editor in chief as well as to the authors to have dictionary entries in the SIDict that are uniform in their form as well as content. In the following we shall illustrate this by two concordances made by the program GREP. In case of the first, we show a section of a dictionary entry classified as "private" [=(biz)], while in the second we demonstrate "idioms" quoted in the dictionary entries. The code for idioms is <66> appearing as * on the printer.

EXAMP.001

nyakleves fn (biz)
Kapott egy ~t, a nyakára v. tarkójára mért ütést.
nyakra—főre hsz
(biz) ~ *hívogat:* újra meg újra.
nyargal ige *~ni*
2. (biz) *Hova ~sz* (= rohansz, szaladsz) *sebesen?*
nyavalyog ige *~ni* (biz)
1. *Tavaly sokat ~tam,* betegeskedtem.
nyel ige *~ni*
4. (biz) *A kocsi ~i a kilométereket,* gyorsan halad.

EXAMP.002

hó\1 fn *havak, havat, hava*
* *Fehér, mint a ~:* tiszta fehér.
holló fn *~k, ~t, ~ja*
* *Ritka, mint a fehér ~:* nagyon ritka.
holt II. fn *~ak, ~at, ~ja*
* *Nem volt se ~, se eleven:* nagyon megijedt.
homlok fn *~ok, ~ot, ~a*
* *Nincs a ~ára írva:* nem látszik rajta.
homok fn -, *~ot, ~ja*
* *~ba dugja a fejét:* nem hajlandó tudomást venni a veszélyről.
* *~ra épít(i terveit, elméletét):* bizonytalan alapra.

4. SIDict as a dictionary data base: DATABASE II

Through the computerized typographical preparation the text body of SIDict was placed into a computer data base called DATABASE I. However, we were compelled to carry out two significant modifications on the data base created during the typographical preparation of SIDict in order to get a computerized version of the dictionary that suits also

the purposes of teaching and linguistic research. Thus by modifying DATABASE I we developed DATABASE II.

Before placing the text of SIDict written in DATABASE I into DATABASE II, we had to carry out the following two modifications:

1) Modification (changing the tilde)

Within DATABASE II it is no longer practical to have the entry heading substituted by the tilde. It is one of the characteristics of the Hungarian language that the last vowel of some stem words are changed when a suffix is added to the stem. In such cases we use tilde with a comma for denoting the stem variant of the suffixed dictionary entry e.g.

```
gólya    (+k,+t,+a) = gólyák, gólyát, gályája       = ~k, ~t, ~ja
gondola  (+k,+t,+a) = gondolák, gondolát, gondolája = ~k, ~t, ~ja
görbe    (+k,+t,+a) = görbék, görbéket, görbéje     = ~k, ~t, ~je
ige      (+k,+t,+a) = igék, igéket, igéje           = ~k, ~t, ~je
```

The "tilde" and the "tilde with a comma" was changed into the lexical entry i.e. its appropriate form variant after the elaboration of the algorhytm by a program written in C language.

2) Modification (morphological analysis)

It seems to be practical to place the dictionary corpus of SIDict DATABASE II that has been morphologically analyzed and lemmatized. This is due, on the one hand, to the fact that the Hungarian language has an agglutinative structure and, on the other to the stem variation mentioned above. The morphological analysis and lemmatization was carried out by the program HUMOR developed by László Tihanyi and Gábor Proszéky.

In the forthcoming we demonstrate this by the showing the previous detail analyzed by the program HUMOR:

```
<5><41>híre-hamva [FN]<40> %szragos %fn (csak [HA] esz
[FN] 3. személy [FN] ·ben [INE], %ált. alanyeset [FN]·
ben [INE]) <4><2>a [DET] tegnap [FN] & le [IK]· esik
[IGE]≈es· ett [Me3]& hó [FN]· nak [DAT] ma [FN]& már [HA]
híre-hanva sincs [IGE], <1>nyom [FN]· a [PSe3] sincs
[IGE], eltűnt [MN]&.

<5><41>híres [MN]<40>
<4><3>I. [SZN]&. <1>%mn <2>híres%ek, híres%t <1>%v.
<2>híres%et, híres%en
<3>1. <2><52> ember [FN], író [FN], politikus [FN]&,
színész [FN]: <1>ki [NM]· emelkedő [FN]
tulajdonság [FN]·ai [PSe3i]·ról [DEL], teljesítmény
[FN]·é [PSe3]·ről [DEL]& jól [HA] ismert [MN]& (=
hírneves [MN]). <2><52> regény [FN], dal [FN], város
[FN]: <1>közismert [MN], nevezetes [MN]. (biz [HA])
<2>Nem [HA]& valami [NM]& <52> ez [NM] a [DET] dolgozat
[FN], <1>gyenge [MN] minőség [FN]·ű [UKEP], középszerű
```

```
[MN]. <3>2. <1>(%pejor) <2>Megint [HA] itt [HA] van
[IGE]·nak [t3] a [DET] híres, <1>rossz [MN] hír [FN]·ű
[UKEP] (= hírhedt [MN]) <2>barát [FN]·aid [PSe2i].
<3>%II. <1>%fn <2>híres%ek, híres%et, híres e [NM]&
<1>(%pejor) <1>Kétes [MN] hír [FN]·ű [UKEP] személy [FN].
<2>Hol [HA] az [DET]& a [DET] híres? <1>(Megszólítás
[FN]·ban [INE]:) <2>No [ISZ], te [NM] híres!

<5><41>híresség [FN]<40> %fn <2>híresség%ek, híresség%et,
híresség%e
<4><3>1. <2> apa [FN]≈Ap·já [PSe3]·nak [DAT] híressége
[NM]&, <1>híres [MN] volta [FN]& <2> nemcsak [KOT] segít
[IGE]·ette [TMe3]&, akadályoz [IGE]·ta [TMe3]& is [KOT]
pálya [FN]≈pályá·já [PSe3]·n [SUP]. <3>2. <2>Az [DET]&
előadás [FN]·ra [SUB] meg [IK]·hív [IGE]·ták [TMt3]& az
[DET]& összes [MN] híresség%et, <1>híres [MN] ember
[FN]·t [ACC].
```

Presently we are currently evaluating retrieval programs that could be used the to query dictionary DATABASE II. The simplest solution would be to use GREP, however we are still making some experiments with programs like KAYE and WordCruncher which serve more purpose and thus can be used in a more comfortable way.

As the dictionary data base has already been analysed morphologically and is full of codes, therefore it is necessary the filter the output resulting from the individual searches through a special program filtering out the codes and transforming the text into a form similar to that of a "normal dictionary".

5. Analysing the vocabulary of SIDict

We want to carry out the analysis of the SIDict vocabulary, on the basis of DATABASE II, in two directions:

5.1. The relationship between lexical entries and words occuring in the dictionary (lexemes):

We want to explore how the 14.000 envisaged lexical entries relate to the words occuring in the the text body od SIDict. Points of view for the research:

a) Are there any words in the text body that are not lexical entries?

b) Are there any lexical entries that occur only in their own lexical record?

Of course we analyse the vocabulary of definitions and sentence examples within the text body separately.

5.2. Classifying the lexical entries and their meaning nuances:

SIDict is made for pupils and, according to the plans, contains words of the basic stoek of vocabulary. This is the reason why we think it worth examining what is the proportion of so-called "classified" lexical entries in comparison to the 14.000 lexical entries and what proportion of the meaning nuances of the lexical entries was classified. Lexical entries classified in the dictionary are to be found on the margin of the base vocabulary. (The classifications were the following: (nép) [= dialectal], (biz) [= family usage, colloquial],

(pej) [= pejorative], (rég) [= archaic], (tréf) [= jocular], (vál) [= refined]). The results of these examinations will yield objective data as regards the size of a planned basic vocabulary treasury. It seems to be most likely that the size estimated by Júlia Pajzs (about 10.000 to 12.000 lexical entries) will be justified (PAJZS 1991).

With the computer-stored data base of SIDict, lexicographers, linguists and those who are interested in such topics will have for the first time a computational explaining dictionary of Hungarian. Due to the students's dictionary character of SIDict we think it can be utilized excellently for educational purposes. The experiences gained while preparing it will be extremely useful in the future when working on more voluminous computational dictionaries e. g. *A magyar irodalmi és köznyelv nagyszótára (1533—1990)* [= the Great Literary Dictionary of Hungarian] (KISS — PAJZS, PAJZS 1988, PAPP — HEXENDORF).

Bibliography:

BÍRÓ ÁGNES: *A Képes Diákszótár módszertani kérdései* [= Methodological questions in connection with the Students' Picture Dictionary]. Az I. magyar alkalmazott nyelvészeti konferencia, Nyíregyháza 1991. május 3—4. Előadások: 56—61. oldal.

KISS GÁBOR: *A Word Perfect szövegszerkesztő programozási lehetőségeinek felhasználása szövegek szótárszerű feldolgozásának előkészítésében — Bemutatva a Vizsolyi Biblia négy Evangéliumán* [= Utilizing the programming possibilities of Word Perfect when preparing dictionary-like elaboration of texts - illustrated on the Four Gospels of the Bible of Vizsoly]. Az I. magyar alkalmazott nyelvészeti konferencia, Nyíregyháza, 1991. május 3—4. Előadások: 392—404.

KISS LAJOS — PAJZS JÚLIA: *A magyar irodalmi és köznyelv nagyszótára (1533—1990)* [= The Great Dictionary of Hungarian Literary and Everyday Language (1533—1990)]. Magyar Nyelv 1989. LXXXV. évf. 2. szám. 129—136. oldal.

LAROUSSE DÉBUTANTS, 20.000 Mots, Direction de René LAGANE, Libaire Larousse, 1986.

OXFORD BASIC ENGLISH DICTIONARY. Edited by Shirley Burridge, Oxford University Press 1981.

PAJZS JÚLIA: *Creating a Historical Dictionary of Hungarian with the Aid of Computer.* BudaLEX '88. Proceedings, Papers from the EURALEX 3rd International Congress, Budapest, 4—9 September 1988. T. Magay and J. Zigány (eds.). p. 559—563.

PAJZS JÚLIA: *A Debreceni Tezaurusz egyik felhasználási lehetőségéről: a magyar nyelv számítógépes alapszókincstára* [One of the possibilities to utilize the Debrecen Thesaurus: the basic vocabulary collection of Hungarian on computers]. Könyv Papp Ferencnek, Tanulmánygyűjtemény Papp Ferenc 60. születésnapjára. Szerkesztette: Hunyadi László, Klaudy Kinga, Lengyel Zsolt, Székely Gábor. Kossuth Lajos Tudományegyetem Debrecen, 1991. 343—348.

PAPP FERENC — HEXENDORF EDIT: *Magyar szókincs a könyvnyomtatástól napjainkig — számítógépre tervezve* [= Hungarian vocabulary since the invention of book-printing till the present - designed for computers]. Magyar Tudomány 1985. XXX. évf. 1. szám 36—40. oldal.

Computational Lexicography in Prague

JAN KRÁLIK

A short survey of computational lexicography in Prague. Main historical and actual projects are mentioned and briefly described.

The Prague Linguistic School underlined the phenomena of potentional linguistic events, quantification or quantitative confrontations *(Mathesius, 1911)* which could neither be fully appreciated nor scientifically exhausted and completely linguistically covered at the time of their definitions in the 1920s. Nobody knew about computers and hardly would even anybody dream about the use of computers in lexicography. Thanks to the traditional exact classification and quantification of linguistic events, though, the first mechanic machines were installed very early in Prague to be applied in linguistic research. This technical equipment was very close to computers and proved most capable of the lexicographic work as well. That is why computational lexicography has deep roots in Prague and its projects are frequent, even if limited by lack of both the hardware and software.

The first experiences with mechanical lexicographic research were gathered by J.Štindlová in the Czech Language Institute in Prague in the late 1950s *(Štindlová, 1961)*. Even if the mechanical equipment applied was confined to only a part of the technology then available (the punch cards machines) and no direct reference to electronic computer was made, the computational lexicographic research of the Czech Language Institute was remarkably successful, starting from that period. Mechanical as well as computational algorithms of data arrangement and selection were worked out and new experiments such as first terminological corpora, author concordance indexes, etc., opened the paths to future projects.

One of the highest projects finished computationally transcribed and indexed the total list of lexicographic items for the Dictionary of Standard Czech (SSJČ), and the lists of nouns and verbs, the reverse alphabetical dictionary, and other lists were obtained. Unfortunately, hardly anything of this work was published, and the fast development of computational technology and some other reasons made those resources incompatible with the advanced computers of the present day.

Soon afterwards, however, in the early 1970s, tradition and a considerable

experience inspired another extensive work, using more advanced means so that its result *is* compatible with the present computers. The Department of Mathematical Linguistics lead by M.Těšitelová started the project of a complex quantitative analysis of modern Czech. The corpora included 180 texts with 3.000 running words each (in total 540.000 words), and every word was supplied with a full morphological and a brief syntactic information. The data were punched on cards and transferred to magnetic tapes and processed by means of old big computers with the help of special programs accounting for each step of quantitative analysis *(Králík, 1982)*. All this data are preserved and were recently transferred transferred into a current computer readable form on floppy discs.

The analysis followed both the quantitative and the lexicographic aspects. The frequency dictionaries of various registers and the whole corpus *(Těšitelová 1983, 1985)* as well as the Reverse Alphabetical Dictionary *(Těšitelová - Petr - Králík, 1986)* were published. Though not lexicographically targeted, thanks to computers, the project gathered a great amount of lexicographically relevant data, including those for a semantic frequency dictionary, which still brings it to the attention of specialists nowadays.

For example, the Czechoslovak press-agency ČSTK used this corpus for the automatic search in Czech newspaper texts which - due to Czech inflection variating suffixes and changing word roots - was a far from easy task. Another project, performed at the Prague Technical University, used the corpora and their computational outputs to support an automatic analysis of spoken Czech, looking for the possible sound and letter combinations in spoken texts (allowed by a relatively close proximity of spoken and written Czech). The contextual lexicographic information as well as described inflectic proved the corpora significant for both the design and verification of projects on Czech spelling-checkers, etc.

The work inspired several specialized lexicographic projects and the research using similar computational technology, serving as a new means of research and influencing the formulation of new projects applying more advanced equipment.

The first of those works, as performed in the Czech Language Institute in Prague, was the Full List of Minor Place Names in Bohemia (including the names of fields, forests, etc.). In this project, headed by M.Knappová, the computers formed the reverse alphabetical list and selected numerous samples. The minor-place names of Bohemia were treated as to their origin and development, both accounting for linguistic and extralingual facts, and communicative function.

A completely new edition of the One-Volume Dictionary of Czech implements the data base system and its floppy-disc version is to be issued. The user software should enable direct and conditioned search and provide the user with additional comfort. This project is coordinated by J.Filipec and V.Mejstřík.

Another example is the Valency Dictionary, which completes the research into the sentence patterns of 1.000 most frequent Czech verbs. The dictionary compiles relevant meanings and valency characteristics, and every verb is described from the morphological, syntactic and semantic points of view including the selective tendencies for filling all valency positions in Czech sentence. Apart from providing various

sorting, search or concordance algorithms, the computer also serves as a desk top publisher. The project is headed by N.Svozilová.

Another recent project is the full lexical corpus for the new computer edition of the Czech Spelling Rules which will offer an essential lexicographic information to the public. The floppy-disc issue including the service programmes and its computational confrontation with new lexicographic corpora are considered for preparation. The project is headed by O.Martincová.

Parallelly to the research of the Czech Language Institute, a highly specialized project has been formulated by V.Smetáček dealing with lexicographic and textual meanings of Czech words. The project introduces the hierarchy of meaning levels and encodes several hundreds thousand meanings and applies the programmes for similarity and semantic likeliness. The work is an original terminological basis and can be easily reordered semiotically *(Smetáček, 1984-1986).*

To the contrary of the past era, when any computational application of in lexicography neared a dream or was exceptional, all the latter projects are technically limited and their introduction is extremely belated, which also holds for a project of the Computational Fund of Czech *(Čermák - Králík - Pala, 1992).*

Despite the present availability of 2 PC only and despite largely limited software (without the Word-Cruncher, e.g.), the Lexicographic Department, headed by F. Čermák, accepted all the offers of textual data by publishing houses. The offer is faster than the Department's possibilities. Even if there are the FD copies of a great amount of textual data offered by newspapers and periodicals (such as Lidové noviny, Mladá fronta Dnes, or Mladý svět), the Department still awaits a project ensured for their balanced selection. Nonetheless, very shortly after the first PC had been installed, the Lexicographic Department formed an original database and started to construct own programs such as for concordances or contexts, etc.) *(Králík, 1992).*

New technology no doubt brings limitations to traditional lexicographic methods. On the other hand, however, it also brings very useful and precise demands and performs enormous amount of work beyond human ability before. New technology always inspires new projects and lexicography is no exception. If a full compatibility is guaranteed, the computational technology is a true revolution for any lexicographer.

References

Čermák, F. - Králík, J. - Pala, K.: Počítačová lexikografie a čeština (Computational Lexicography and Czech). Slovo a slovesnost 53, 1992, No.1, pp.41-48.

Kirschner, Z.: MOSAIC - A Method of Automatic Extraction of Significant Terms from Texts. Praha 1983.

Králík, J.: Technika zpracování hromadných dat (Data Processing Technology). In: M.Těšitelová et al.: Kvantitativní charakteristiky současné české publicistiky (Quantitative Characteristics of Present-Day Czech Journalism). Linguistica II, ÚJČ ČSAV, Praha 1982, pp.72-80.

Králík, J.: Kapitoly o výpočetní technice / k problémům komunikace lingvista - programátor - počítač (Chapters on Computers / Some problems of communication: linguist - programmator - computer). ÚJČ ČSAV, Praha 1987.

Králík, J.: Czech Language and Computers. Czechoslovak Life 1992, 4, p.24-25.

Pala, K. - Osolsobě, K.: Czech Stem Dictionary for IBM PC XT/AT. In: Conference on Computational Lexicography. Balatonfüred, September 1990.

Panevová, J. et al.: Lexical Input Data for Experiments with Czech. Praha 1981.

Smetáček, V.: Automatizovaná báze lexikálních jednotek BALEX (The Automatized Base of Lexical Units BALEX). Internal print ÚVTEI, Praha 1984-1986.

Štindlová, J.: Stroje na zpracování informací a jejich význam pro jazykovědu (Machines for Information Processing and Their Importance in Linguistics). Slovo a slovesnost 22, 1961, No.3, pp.208-215.

Těšitelová, M. et al.: Frekvenční slovník češtiny věcného stylu (Czech Frequency Dictionary of Journalism and Science). Praha 1983. Internal print ÚJČ.

Těšitelová, M.: Kvantitativní charakteristiky současné češtiny (Quantitative Characteristics of Present-Day Czech). Academia, Praha 1985.

Těšitelová, M. - Petr, J. - Králík, J.: Retrográdní slovník současné češtiny (Reversed Alphabetic Dictionary of Czech). Academia, Praha 1986.

Towards a Computerized Historical Dictionary of Dutch: from Printed Dictionary to Correct Text File

J.G. KRUYT — J.J. VAN DER VOORT VAN DER KLEIJ

The Woordenboek der Nederlandsche Taal, the Dutch counterpart of the Oxford English Dictionary, is subject to computerization. Presently, the method of Optical Character Recognition (OCR) is applied for converting volumes that are available in printed form only. The correction of the OCR-output files and the administration of the correction process is computerized as much as possible. An automatic preprocessing improves the OCR-output file in order to reduce and facilitate the computer-aided manual correction. In spite of extensive correction still being required, the present method has resulted in a considerable reduction of costs and an improved efficiency in the organisation of the project compared to the earlier method of word processing by a commercial firm.

1. Towards a computerized historical dictionary of Dutch.

The Woordenboek der Nederlandsche Taal WNT is a monolingual dictionary based on historical principles, describing the Dutch vocabulary from 1500 up to the 20th century. The first fascicle was published in 1864. The dictionary will be completed in 1998. It will then comprise more than 100.000 colums of dictionary text. The WNT can be considered the Dutch counterpart of the Oxford English Dictionary OED, the Deutsches Wörterbuch started by the Grimm brothers, and the Dictionary of the Swedish Academy (SAOB).

The WNT is currently being computerized. The Electronic WNT will be a text file encoded for information categories in SGML-format (Bryan 1988), similar to the electronic New OED (cf. Kazman 1986). That is, the running text will be interrupted by codes specifying the type of information a text fragment conveys (cf. De Bruin et al. 1991). A first prerequisite is a machine-readable version of the dictionary text. Since 1982, text processing facilities have been utilized in the production of the printed dictionary, which at the same time results in corrected machine-readable fascicles. The ca. 69.000 colums of dictionary text produced before 1982, however, are available in printed form only and have to be converted to machine-readable form.

Until recently, a commercial firm was keying-in the text using word processors, and the output was manually corrected. Not only the proper text was keyed-in, but also some textual adaptations supporting the future automatic encoding of information categories (cf. Van der Voort van der Kleij & Kruyt 1992: section 1). These adaptations had previously been noted by WNT-lexicographers onto enlarged copies of the dictionary pages. Except for some minor points, this method is essentially similar to the one applied in the New-OED project (cf. Simpson 1986, Kazman 1986, Berg et al. 1988).

At the very end of 1990, the application of OCR was considered as an alternative for keying-in, mainly for financial reasons. After extensive experiments, a Kurzweil K5200 OCR system has been decided on, in spite of the rather poor quality of the output. In contrast with the former situation, institutional personnel, rather than a commercial firm, was envisaged to be charged with the conversion of the printed text into a corrected machine-readable version. Making use of the computer facilities at our institute, a system has been developed by which the conversion and correction process and its administration are computerized to a major extent. The description of this system will be preceded by a concise discussion of some characteristics of the printed dictionary complicating the conversion by OCR.

2. The printed dictionary: some features complicating OCR.

Several characteristics of the printed dictionary text complicate the conversion by OCR. Like the OED (Berg et al. 1988), the uneven quality of the print from the old metal plates and the complex grahical structure cause problems. Print quality is far from optimal due to different shapes of identical characters, broken characters, connected characters that should be separate, uneven brightness etc. Additionally, more than ten different print types have been used in the 504 fascicles printed before 1982.

Graphical structure is characterized by numerous type fonts, numerous special characters and symbols, and various indentation sizes which reflect the hierarchical structure within the entry. Apart from the more usual type fonts roman, italics and bold face, the WNT includes small capitals, spaced roman, spaced italics, spaced small capitals, and in the older volumes additionally a smaller variant of small capitals. Moreover, type fonts alternate rather frequently. Special characters particularly occur in the etymology sections and in quotations

derived from regional sources (transcribed dialect). Special symbols concern symbols in quotations (such as mathematical symbols, and the old Dutch pound symbol), as well as some functional marks in the dictionary text (such as the two vertical lines separating the lexicographer's text from the illustrating quotations). In the older volumes, graphical structure is moreover complicated by centred quotations derived from poetry. The structural indentations are relevant to the discrimination of thirteen potential levels in sense hierarchy (cf. De Bruin et al. 1991).

As for the future automatic encoding of information categories, formal characteristics are essential to the automatic assessment of contents (cf. Kazman 1986). Graphical characteristics need therefore to be preserved in the machine-readable text file. For that reason, the effects of the abovementioned factors have been investigated by intensive experimentation. The results indicated that extensive correction would be required, that different OCR training sets (cf. 3.1.) would have to be developed for the various parts of the dictionary, and that it is not sure whether the old volumes including centred quotations can be tackled by scanning/OCR. In spite of these consequences, this method was attractive to our project mainly for financial reasons. Subsequently, it proved to result in an improved efficiency in the organisation of the project (cf. 4.).

3. From printed text to correct text file.

Main components of the process from printed text to correct text file are scanning and optical character recognition, an automatic preprocessing in order to improve the OCR-output, and computer-aided manual correction at a computer screen.

3.1. Scanning and OCR.

A Kurzweil K5200 connected with a Commodore 386SX-16 personal computer is used for scanning and OCR. The first stage is the development of a training-set. A number of pages is scanned, and, during scanning, all characters that are not unambiguous to the system are presented at the screen. Only characters that have a correct and claercut shape are confirmed to the system as being correct. In this way, the system specifications for each character are refined according to the specific characteristics of the text to be processed. If structurally relevant special symbols cannot correctly be processed by the Kurzweil, a simulation is trained. For example, the two vertical lines " ‖ " indicating the beginning of a quotation block is trained as "!)!)", and is automatically converted into the original symbol in the automatic preprocessing phase (3.2.). This also applies to the dash "—", which is trained as "+)". The training-set is stored and can essentially be used for all pages with similar textual characteristics. For our dictionary, the necessity of developing different training-sets because of the various print types, has already been referred to in section 2.

After the training phase, portions of 16 columns of dictionary text are scanned with use of a form feeder, subsequently recognized, and then converted to a suitable format, in our case an ASCII-file. The portion of 16 columns, each column containing about 3500 characters on the average, proved to be a manageable quantity for computer-aided manual correction (3.3.). Training, scanning, character recognition, and conversion require 2 minutes per column on the average.

The output is a running text file interrupted by codes, as shown by the following examples:

>>>UI, <8,R>Samenst. enz. <12,R>34

<8,R>grootte uitgezocht, <6,R> REINDERS, <8, I>Landb <8,R>2,201 [1893]. Uienzaad wordt in ons land ingevoerd voor het

...

>>><8,I>+)>>>Uienzweefvlieg <8,R> (3), zie de aanh. !)!) Kleine narcisvheg of uienzweefvlieg <8,I> (Euvierus strigatus),

(WNT, Vol. 17-3, column 34).

The codes indicate the beginning of roman ("R") and italic ("I") font only, as well as the relative height of following characters ("12", "8", "6", and other values). Note that codes only mark font shifts and height alterations. Many features of the printed dictionary (other fonts, special characters and symbols) cannot be represented in the OCR-text file. Note, however, the successful simulations "!)!)" and "+)".

The quality of the OCR output text file is improved by automatic preprocessing (3.2.), in order to reduce and facilitate the computer-aided manual correction at the screen (3.3.).

3.2. Automatic preprocessing.

Before the OCR output text file is copied from the PC to the central hostcomputer Digital VAX 8350, the PC non-ASCII values are converted to VAX non-ASCII values. The text file is now characterized by the extension ".ICR". i.e. "scanned and processed by the Kurzweil".

Every night, the presence of OCR text files is automatically checked in a batch procedure, on the basis of the filename extension. If present, OCR text files are converted into VAX/VMS format and subsequently processed by a program written in VAX-BASIC, that improves the proper text by controlling as many as possible systematic features. Essentially two types of operations are involved. Some concern the correction of errors consistently made in the OCR process, such as "DI" being corrected into "Dl", and "(11)" into "(II)". Focus, however, is on the correction of structurally relevant strings, such as correcting "QCU– " into "– QCU", "QCU" being the code indicating the beginning of italic font (cf. below).

Other operations concern structural improvements, rather than the correction of consistent errors. They include the insertion of fonts not, or not properly represented in the OCR text file. For example, for each text file small capital font is calculated on the basis of the digits in the OCR-codings (like those shown in 3.1.) and specific contextual features. Bold face is determined on the basis of either pattern recognition or a unique combination of textual characteristics. Roman font indications are removed, being considered the default font. Another operation concerns font code completion. The graphical codes for italic font, bold face and small capital font are completed by end codes. For example, "QCU" means "start italic font", "ZCU" representing "end italic font". Special symbols, such as the two vertical lines and the dash (cf. 3.1.), are reconstructed. Structural indentations are checked on correct location and visualized by five subsequent copyright symbols. Running headlines are not yet removed in order to support manual correction, but standardized so as to be able to remove them automatically in a later stage of the computerization process. Words broken up at the end of a line by a hyphen are automatically connected. The number of spaces is regulated. Although different types of operations are involved in the preprocessing

process, most rules in the program are based on textual patterning.

For an impression of the effect of the automatic preprocessing, in particular of the structural improvements, the example lines shown in 3.1. are presented here after they have been processed by the program.

> *#$ KOPREGEL *#$ UI, Samenst. enz. 34
>
> grootte uitgezocht, QKKREINDERSZKK, QCULandb.ZCU 2, 201 [1893]. Uienzaad wordt in ons land ingevoerd voor het
>
> ...

©©©©© – QCUUienzweefvliegZCU (3), zie de aanh. // Kleine narcisvheg of uienzweefvlieg (QCUEuvierus strigatusZCU),

The resulting text file has more or less the layout of the printed dictionary. Depending on the quality of the OCR text file, the preprocessing introduces 2000-3000 modifications in a file of sixteen dictionary columns. Because of the inconsistency discussed in section 2, some components of the preprocessing program will have to be adapted if textual characteristics change.

The automatically preprocessed file is characterized by the extension ".conv", which means "converted and automatically preprocessed text file". This file is input for the computer-aided manual correction.

3.3. Computer-aided correction.

Correction comprises three phases. The first one is focussed on correction of the entire text, the second one particularly on textual features that are of primary importance to the future automatic text encoding in terms of information categories. First and second correction are performed by use of written instructions. The third phase concerns a final check by the present authors. Standard for correction is the printed dictionary text. Errors in the original text are not corrected, unless the error concerns an element essential to the future automatic encoding of information categories (cf. 4).

Correction is performed at the screen by use of VAX-terminals. A text file is interactively assigned to a particular corrector, either for first or second correction. Each time a file has been read from disk and appears at the terminal, some automatic operations improve structural consistency. This concerns, for example, the regulation of spaces in quotation dates. For ease of correction, text structure is clarified by addition of screen attributes (reverse video, highlighting etc.) to the font codes. During correction, four buffers are visualized as windows at the screen: a text buffer including the text to be corrected, a comment buffer for comments by the corrector, a message buffer which informs the corrector about proper system messages as well as messages concerning correctness of specific textual features, and a command prompt buffer. This is essentially similar to the system described in Van der Voort van der Kleij & Kruyt (1992: section 2), except for two major differences: screen attributes do not replace font codes as the latter may be subject to correction, and corrections can freely be made in the text buffer.

Correction facilities include a menu supporting the insertion or modification of font codes. Specially defined keys support frequent operations, such as inserting or revising structural indentations, and connecting words broken up at the end of a line in cases where the

hyphen has not been detected in the OCR process. In the second correction phase, predefined keys support sophisticated searches in order to check all aspects essential to the automatic encoding of information categrories, and frequently occurring errors originating in the OCR-phase that could not be corrected by automatic preprocessing. All these means aim at an optimal check on correctness of textual features during the first and second correction process. When a text file is completely corrected, either in the first or second correction phase, and the final session is finished, formal aspects of font codes are automatically checked and interactively corrected, because of their relevance to following stages in the project.

The final check and correction is facilitated by a connection between the line numbers in the text buffer containing the corrected dictionary text file and those in the comment buffer containing the second corrector's comments. In this way, scrolling and searching is reduced to a minimum. For a more detailed description of a similar facility, we refer to Van der Voort van der Kleij & Kruyt (1992: section 3).

The procedures described here are mainly based on textual patterning and are realized by use of the extendable VAX-editor EVE in combination with the programming language Text Processing Utility (TPU). First, second and final correction phase of each file is characterized by the file extension (".conv_cor", ".conv_cor_cor" and ".conv_cor_def", respectively). First correction requires 32 minutes per column on the average. The second and final correction procedures have recently been developed and tested. Total correction duration is estimated at ca. 45 minutes per column on the average.

3.4. Administration.

The administration of the process described above is computerized to a major extent. Administrative data are automatically recorded in a separate file (sequential fixed file), by a VAX-BASIC program using input-files produced by the editing program.

Before scanning, filenames are constructed by a program. An example of a filename is "17S30001", specifying the volume number (17-3) and the number of the first out of sixteen columns in a text file (0001). These filenames are included in the administration file. Subsequently, extensions specifying the phase of a text file in the whole process of computerization are added to the filenames. After OCR, the filename is provided with the extension ".ICR" (cf. 3.1.). After the automatic preprocessing, changing the extension into ".conv" (cf. 3.2.), a marker is automatically inserted before the filename in the administration file, indicating that the file is ready for computer-aided manual correction. Files to be corrected are interactively assigned to correctors. During correction, the various phases are recorded in the batch by use of the standard time provisions of the computer system, including the date and time of the beginning and end of the first and second correction, respectively. Correction durations can easily be computed. When first, second and final correction have been finished, the file extension is changed.

This way of administration not only ensures a correct overview of the files in their various phases, but also facilitates planning the project by use of rather objective durations of operations.

4. Discussion.

In the New OED-project, OCR was rejected because of the poor results (Berg et al. 1988). Malmgren (1988:167) on the other hand, reports with respect to the Dictionary of the

Swedish Academy, that OCR (by a Kurzweil) must be followed by "some manual correction". In our dictionary project, extensive correction is required. The different experiences confirm that OCR is still a controversial tool for computerizing large scale lexicographical resources. However, the feasibility of OCR may not be considered from the point of view of correction only. Several other aspects may be relevant in the judgement on OCR being a feasible approach.

To our project, the method presented in this paper has substantial advantages over the earlier method of word processing and correction performed by a commercial firm. First of all, a considerable reduction of costs per column has been achieved. Another major result concerns the organisation of the project. The production of machine-readable dictionary text by both OCR and word processing (new fascicles) at the institute implied a reconsideration of the various stages in the computerization of the dictionary. The textual adaptations required for the future automatic encoding of information categories (cf. 1) preferably applied to all machine-readable text, irrespective of the origin. In contrast with the former method (cf. 1), this textual adaptation is presently performed after the text has been made machine-readable. This activity has been computerized as well, which resulted in a considerable saving of time and an improved quality (cf. Van der Voort van der Kleij & Kruyt 1992). Additionally, the availability of many text files stored on disks of the institutional computer supports research into textual features, allowing lexicographical hypotheses to be tested and textual patterns unknown so far to be revealed. This research has been relevant to the development of the computer programs for the automatic preprocessing and the computer-aided correction described above, and is furthermore particularly relevant to the automatic encoding of information categories. The system described above also allows correctors working at home, using telecommunication facilities, without any essential change of the procedures. In conclusion, apart from the computerized administration (3.4.) and the reusability of developed procedures and programs implied above (3.3.), these factors contribute to an improved efficiency in the project and an optimal quality of the product.

These types of improvements are of prime relevance for the feasibility of a large scale project like ours. The question therefore is whether the correction of OCR text files could be combined with simultaneous linguistic structuring as suggested by Malmgren (1988), with the correction of errors in the original dictionary text (cf. 3.3.), or with improving internal consistency. Our experience is that this is feasible to a limited extent only. The instructions grow too complicated. Errors in the original text often require extensive looking up procedures. Improvement of consistency is preferably performed by consistent automatic procedures rather than by inconsistent human correctors. For these reasons, we have chosen to postpone this kind of improvements to a later stage in the project.

Our conclusion is that applying OCR should be considered as an integrated component of the computerization process as a whole, rather than by the quality of the immediate output by itself. Although needs and conditions may differ from those of our project, aspects of our approach may be useful to other projects as well, in particular those concerned with complicated structured text.

References.

Berg, D.L., Gonnet, G.H. & Tompa, F.Wm. (1988), The New Oxford English Dictionary Project at the University of Waterloo, UW Centre for the New Oxford English Dictionary, OED-88-01.

Bryan, M. (1988), SGML, An Author's Guide to the Standard Generalized Markup

Language. Addison-Weshley Publishing Company, London.

De Bruin, H.J.B.A., Van der Voort van der Kleij, J.J.W. & Kruyt, J.G. (1991), Algoritmen voor een dictionary entry pareser voor het elektronisch WNT, INL Working Papers 91-02.

Kazman, R. (1986), Structuring the Text of the Oxford English Dictionary through Finite State Transduction, doctoral dissertation University of Waterloo, Data Structuring Group CS-86-20.

Malmgren, S.-G. (1988), The OSA project: Computerisation of the Dictionary of the Swedish Academy. In: Literary and Linguistic Computing 3, 166-168.

Simpson, J. (1986), Opening Address: The New OED Project. In: Information in data. Proceedings of the First Conference of the UW Centre for the New Oxford English Dictionary, 1-6.

Van der Voort van der Kleij, J.J. & Kruyt, J.G. (1992), Restricted editing in a corrected dictionary text file, Proceedings COMPLEX'92.

Acknowledgements.

Mrs. M.L.W. van Bennekom and Mr. R.J. van Strien substantially assisted in the experiments (cf. 1). Mrs. Van Bennekom additionally contributed to this paper by her experience with the OCR-system.

Phonetic Syllables in French: Combinatorics, Structure and Formal Definitions

ERIC LAPORTE

Abstract

The notion of syllable is based on particularly clear intuitions and is relevant to all languages, but no good formal definition is known. One of the problems is that an utterance can be divided into syllables in several ways, e.g. `[a/tlas]` and `[at/las]` for *atlas* in French. Different algorithms can be designed to syllabify phonetic strings or to divide them into other units. We define, test and discuss criteria of choice between such algorithms.

Syllabication algorithms have been studied for the purpose of hyphenation by computer (e.g. Liang, 1983; Désarménien, 1986; Mañas, 1987), since in many European languages syllabication and hyphenation have much in common. However, for no language they are completely identical: e.g. *atlas,* the French example above, cannot be hyphenated. Our matter here is a linguistic one, so we syllabify phonetic and phonemic strings, regardless of spelling.

1. Intuitive pronounceability

Clear intuitions about syllables are a valuable starting point. One of them is that the existence of syllables has something to do with pronounceability. A phonetic transcription is made of phonetic symbols, but not any sequence of phonetic symbols is a valid phonetic sequence in a given language, it can be unpronounceable or unconceivable in that language. Note that pronounceability is not defined by universal abilities of human vocal tractus, but is relative to a given language: French speakers encounter no difficulty in uttering such sequences as [drwa] (*droit*) and [plɥi] (*pluie*) but are generally puzzled if asked to repeat [ɕwri] in It. *Euridice*. A sequence of phonetic symbols is pronounceable if and only if it is made of pronounceable syllables ordered in pronounceable combinations: e.g. the unattested sequence *[pastjɔ̃d] sounds French. An interesting feature of these intuitions is that they are not restricted to linguists but widespread among most kinds of speakers.

However, these intuitions are too fuzzy to provide a reliable basis to a formal definition of phonetic syllables. We chose to confront them with lexical phonetic data.

Phonetic strings are transcribed with the concern of maximal closeness to utterances, whereas phonemic strings are scholarly constructions. Since the intuitive basis of the notion of syllable is connected to pronounceability, phonetic representatives of utterances seem to be the most adapted material for experiments. Phonetic sequences are written in IPA; however, we apologize for writing uvular *r* as [r].

2. Characterization of pronounceable sequences

A list of words of a language is not a satisfying characterization of the pronounceable sequences in that language, no matter how exhaustive the list is. Phonetic sequences which are clearly not accepted as words of the language can be clearly pronounceable. In order to provide a characterization of pronounceable sequences in French, we will rely on the intuition that such a sequence is a succession of elementary pronounceable phonetic sequences ordered in pronounceable combinations. These elementary sequences may be syllables or other kinds of objects, but they may not be simply the phonetic symbols since consonants are not pronounceable in isolation.

For clarity we will adopt the terminology of mathematical word theory. A finite set of objects is called an alphabet when one considers sequences of them: examples of alphabets are the set of phonetic symbols and the set of all possible syllables. The set of all possible finite sequences of symbols of an alphabet A is noted A^*, and any subset L of A^* is called a formal language on A. The empty sequence is made of no symbol at all and we will note it <E>. With these terms, if A is the alphabet of all phonetic symbols, A^* is the set of all finite sequences of phonetic symbols in any combinations, and the set of pronounceable sequences is a formal language on A, but no formal characterization of it is known, not even the list of its elements, which is too large. A more realistic characterization of pronounceable sequences would take the following form:

- an alphabet S of elementary pronounceable phonetic sequences,
- a formal language T on S.

A word of T would thus be transcribed as a sequence of, on an average, 4 elements of S, instead of a phonetic transcription made of, on an average, 9 phonetic symbols. There is an analogy with Chinese writing where each ideogram is pronounced as a syllable. We investigated several reasonable choices for S:

- (i) syllables, e.g. `[par/ti/ky/la/rism]` for *particularisme*;
- (ii) so-called antisyllables, e.g. `[pa/arti/iky/yla/ari/ism]`;
- (iii) vowel-consonant(s) sequences, e.g. `[p/art/ik/yl/ar/ism]`;
- (iv) consonant(s)-vowel sequences, e.g. `[pa/rti/ky/la/ri/sm]`.

In each case, the list of the elements of S, or any other formal characterization of S, tells the internal structure of these elements. This structure is restricted by some constraints (internal constraints): e.g. some consonant clusters are possible in a language and not in another.

In each case, a formal characterization of T tells how the elements of S can assemble together (combinatorial constraints). This characterization is the simplest in the case of antisyllables: medial antisyllables can be distributed into 169 classes $C_{l,\,r}$ according to their left l and right r vowels, initial antisyllables into 13 classes B_r according to their right vowel, and final antisyllables into 13 classes E_l according to their left vowel; a sequence is in T if and only if:

- it begins with an element of B_r,
- it possibly continues with elements of $C_{l,\,r}$,
- it ends with an element of E_l,
- the element of B_l is followed by an element of $C_{l,\,r}$ or E_l,
- and every element of $C_{l,\,r}$ is followed by an element of $C_{r,\,s}$ or E_r.

The combinatorics of antisyllables is restricted by purely formal constraints which are of no linguistic interest. In contrast, the combinatorics of syllables, that of vowel-consonant(s) sequences (VC^*) and that of consonant(s)-vowel sequences (C^*V) are language-dependent linguistic data (cf. sections 4, 5, 6).

Internal and combinatorial constraints are not completely arbitrary or irregular, but partially consistent with acoustic and articulatory categories of phonetic segments. For example, in a consonant cluster, only the first and the last consonants can be semivowels[1] (`[j]` `[ɥ]` `[w]`): `[ʒoajri]` *joaillerie*, `[bwa]` *bois*. In French, this partial consistency is rather limited: the history of the language brought about distributional gaps, and foreign borrowings introduced rare sequences which distort the symmetry of the system. These borrowings cannot be considered marginal and neglected for the sake of symmetry, since many of them are ancient, frequent and not obviously recognizable. For example, take the following rule: when a consonant cluster begins with `[l]` or `[r]` and contains at least two other full consonants[2], then it contains only two other full consonants and the second one is `[l]` or `[r]`, like in `[filtr]` *filtre*. We extracted from a dictionary the exceptions to this rule. All are borrowings or learned words, but some of them are frequent and easy to pronounce: *absorption, marxiste, perspicace, solstice,*

1. We include semivowels in the list of consonants; we consider glides as semivowels, even between a vowel and a consonant.

2. A full consonant is a consonant which is not a semivowel.

superstitieux... There are 111 exceptions (each verb counts as one; the beginning of the list is given in appendix 1).

Syllabic structure is not only a theoretical problem but also a lexicological one (cf. Aubergé et al., 1988). We made experiments on DELAP-FH[3] (Laporte, 1988), a list of phonetic transcriptions of French inflected words, excluding proper names. Homonyms are transcribed only once and the total number of transcriptions is 218,700. The dictionary contains borrowings, so the experiments take into account exceptional words as well as others. The transcription is narrow, especially if compared to that of publishers' dictionaries. In case of phonetic variants, the dictionary gives one of the variants, e.g. [kole] and not [kɔle] (*coller*), [fnɛtr] and not [fønɛtr] (*fenêtre*). In the latter case, the choice of the shorter variant produces many observable consonant clusters. In the sequel, we do not consider phonetic sequences lapping over two words, like [lɔ̃ŋvy] (*longue vue*). We will thus refer to word-initial sequences as initial, etc. The figures in the paper will obviously be higher when one considers sequences lapping over word limits and new phonetic variants.

We extracted from the dictionary the set *S* in the 4 cases above. The set of syllables (i) is described in section 6. The following table shows the size of *S* in the case of antisyllables (ii), *VC** sequences (iii) and *C*V* sequences (iv). The line for isolated sequences stands for the case when an entire word happens to be an element of *S*: these words are interjections (*brr ! pff ! pfft ! pst !*) and elided words, which are not in DELAP-FH.

	(ii) Antisyllables	(iii) Vowel-consonant(s)	(iv) Consonant(s)-vowel
Initial	1,178	216	1.183
Medial	9,189	2,159	2,483
Final	589	593	138
Isolated	5	5	5
All types	10,961	2,438	2,794

3. Antisyllables

Antisyllable boundaries are located in the center of vowels, i.e. in the most stationary regions of speech. This means that provided the sequence of antisyllables is consistent with the formal constraint above, the acoustic content of an antisyllable is relatively independent of the content of adjacent ones. This property might make antisyllables an interesting unit for speech synthesis and recognition. Unfortunately, they are much more numerous than syllables, *VC** sequences and *C*V* sequences.

Medial antisyllables stretch from a vowel to the next vowel. Initial antisyllables stretch from a word boundary to the first vowel, and final antisyllables from the last vowel to the next word boundary. Consequently, the three sets are pairwise disjoint.

---

3. An acronym for the LADL's phonetic dictionary of inflected forms with homonyms assembled together.

4. Vowel-consonant(s) sequences

To obtain the list of *VC** sequences, we divided each word before each vowel, e.g. [p/art/ik/yl/ar/ism]. For each word, the first sequence obtained is labelled as initial *VC** sequence but contains in fact the initial consonant cluster, here [p], since the first division occurs before the first vowel. We speak of a consonant cluster even in the case of an isolated consonant or of no consonant at all (empty cluster). The other sequences contain a vowel and a consonant cluster. This explains the small number of initial *VC** sequences (216) in comparison with that of final *VC** sequences (593).

We used the list of medial *VC** sequences to study the distributional constraints between a medial consonant cluster and the preceding vowel and to implement a model of them. If the 529 attested medial consonant clusters combined freely with the 13 French vowels, there would be 6,877 medial *VC** sequences, but only 2,159 (31 %) appear in the dictionary. Some interdictions are systematic, e.g. a nasal vowel is never followed by a semivowel; other seem contingent or anecdotal, e.g. *[irʒw] is not attested, though [irʒ] (*virginal*) and [urʒw] (*bourgeois*) are. Note that [irʒw] might be attested in a proper name. We modelled the most systematic constraints in the form of a finite automaton which recognizes 6,357 *VC** sequences out of the theoretical 6,877 (92 %). This automaton is shown in appendix 2.

5. Consonant(s)-vowel sequences

The definition of *C*V* sequences is symmetrical with that of *VC** sequences, e.g. [pa/rti/ky/la/ri/sm]. For each word, the last sequence obtained after division is labelled as final *VC** sequence but contains in fact the final consonant cluster, here [sm], since the last division occurs after the last vowel. The other sequences contain a consonant cluster and a vowel. This explains the small number of final *C*V* sequences (138) in comparison with that of initial *C*V* sequences (1,183).

We used the list of medial *C*V* sequences to make a model of the distributional constraints between a medial consonant cluster and the following vowel. If the 529 attested medial consonant clusters combined freely with the 13 French vowels, there would be 6,877 medial *C*V* sequences, but only 2,483 (36 %) appear in the dictionary. Most interdictions seem little systematic, e.g. medial [ksu] is not attested except in proper names, though [ksy] (*sexuel*) and [kswa] (*aixois*) are. However, two interdictions are systematic: the semivowel [ɥ] is never followed by the vowel [y], and [j] is never followed by [i], except in [nji] (*rejoignit*); this can be implemented in a simple finite automaton which recognizes 6,722 *C*V* sequences out of the theoretical 6,877 (98 %). If we compare the percentages for *C*V* and *VC** sequences, we observe that *VC** sequences are more constrained than *C*V* sequences, and that the systematic constraints are stronger.

6. Syllables

The main problem in syllabication is to locate the syllable boundary when a medial consonant cluster contains several consonants, e.g. to choose between [a/tlas] and [at/las] (*atlas*). In case of an isolated consonant or of a hiatus, syllabication is not controversial, at least in French: [ra/pid] (*rapide*), [na/if] (*naïf*).

In order to discover criteria for locating syllable boundaries, let us return to intuitions. In syllabified speech, we alter the normal utterance by inserting pauses periodically. Notice that this alteration is always possible, for any pronounceable

utterance, and that the resulting intermittent utterance is always pronounceable. This is why the notion of syllable is so general. Now the pause inserted in an utterance always divides a medial consonant cluster in two (possibly empty) consonant clusters. Let us translate the assertion above in terms of consonant clusters: any pronounceable *VC*V* sequence can be divided in a pronounceable final *VC** sequence and a pronounceable initial *C*V* sequence. This is consistent with intuition and means that syllabication is possible. Moreover, a pause may not always be inserted at any point of an utterance: e.g. if we divide [ʒo/a/jri] (*joaillerie*), one of the pieces is unpronounceable (*[jri]).

This suggests a test to check whether a syllabication algorithm is consistent with intuition. If we could check whether a final *VC** sequence and an initial *C*V* sequence are pronounceable, we could thus test a necessary property of syllabication. Unfortunately, pronounceability is not a formal notion. But we can test an approximation of it, if we use a (large enough) dictionary.

Assume that for any pronounceable final *VC** sequence, the final cluster appears as final cluster in at least one word, and assume the same for pronounceable initial *C*V* sequences. Then a good syllabication algorithm should divide any attested medial cluster into an attested final cluster and an attested initial cluster.

Both of the assumptions above are false: e.g. [ɛlts] is pronounceable in [fɛlts/pat] (*feldspath*), but [lts] never occurs at the end of a word; more spectacular, [gje] is perfectly pronounceable in [di/vyl/gje] (*divulguiez*), but no French word begins with [gj]. But we implemented the test anyway and tested 3 syllabication algorithms. The results are consistent with intuition.

6.a. The first algorithm has little to recommend it: *if a medial consonant cluster ends with a semivowel, divide before the semivowel, else divide after the cluster,* e.g. [parl/je] (*parliez*), [filtr/e] (*filtrer*). Once a medial cluster is divided, the right piece is always attested as initial cluster, but in 269 out of the 529 attested medial clusters (51 %), the left piece is not attested as final cluster, e.g. [lr] from [kulr/a] (*coulera*).

6.b. The second algorithm is the one used in our phonemics-to-phonemics conversion algorithm (Laporte, 1990) to determine the aperture of mid vowels: *if a medial consonant cluster begins with a semivowel followed by another consonant, divide after the semivowel; if it begins with another sonant* [l r m n ŋ ɲ] *followed by a full consonant*[4]*, divide after the sonant; else divide before the cluster,* e.g. [ʒo/aj/ri] (*joaillerie*), [par/le] (*parler*), [a/si/ste] (*assister*). Here the left piece is always attested as final cluster, but in 156 out of the attested 529 attested medial clusters (29 %), the right piece is not attested as initial cluster, e.g. [psj] from [o/psjɔ̃] (*option*).

6.c. The third algorithm was specially designed to stand the test. For every medial cluster we extracted from the dictionary all the ways of writing it as the concatenation of an attested final cluster and an attested initial cluster. These data suggested an algorithm which produces an acceptable division in most cases: *if a medial consonant cluster contains one of the symbols* [p t k b d g f v] *followed by one of the symbols* [l r]*, say that these two symbols count as one; divide the cluster before the last symbol which is not a semivowel,* e.g. [ku/ple] (*coupler*), [par/le] (*parler*), [op/te] (*opter*), [de/plwa] (*déploie*), [par/lje] (*parliez*), [op/sjɔ̃] (*option*), [eks/kly] (*exclu*). This algorithm is similar to most syllabication algorithms based on tradition (e.g. Mañas, 1987). Only 22 out of the 529 medial clusters (4 %) are not divided into an

4. A full consonant is a consonant which is not a semivowel.

attested final cluster and an attested initial cluster. Among these 22 clusters, 4 contain the sequence [gj] which is not attested as initial cluster though it is pronounceable in association with most vowels. The other 18 appear in 32 words only. The list is given in appendix 3. For verbs only one of the conjugated forms is given in the list.

The test agrees with intuition to indicate that the three algorithms given above are in order of increasing adequacy.

The following table gives the number of syllables in the case of algorithms **6.b** and **6.c**:

	6.b	**6.c**
Initial	1,589	2,690
Medial	1,783	2,140
Final	5,229	3,845
Isolated	3,886	3,886
All types	7,860	6,972

The line for isolated syllables contains the number of monosyllabic words, which of course is independent of the syllabication algorithm. These figures suggest another criterion to compare syllabication algorithms. The better algorithm, **6.c**, leads to a smaller set of syllables. This is not surprising, but notice that the set *S* is still much larger than in the case of *VC** and *C*V* sequences. In the algorithm **6.b**, 3,886 of the 7,860 syllables (49 %) are attested as monosyllabic words. With the algorithm **6.c**, this proportion rises to 56 %.

Conclusion

As a byproduct of this study, we investigated distributional constraints between several elements of a syllable, e.g. between a medial consonant cluster and the preceding vowel, or between the left and the right parts of a consonant cluster. Up to now, such distributional constraints have been easily expressed in finite automata (cf. appendix 2). It is likely that most of the contraints at syllabic level will fit in a global automaton. If so, we will have a formal model of pronounceability.

References

Aubergé, Véronique, L.-J. Boë, J.P. Lefèvre. 1988. "Lexiques et groupes consonantiques, *17es Journées d'étude sur la parole,* pp. 55-60, Nancy.

Désarménien, Jacques. 1986. "La division par ordinateur des mots français : application à TEX", *Technique et science informatiques,* vol. 5, no. 4, pp. 251-265, Paris: AFCET-Gauthier-Villars.

Laporte, Eric. 1988. *Méthodes algorithmiques et lexicales de phonétisation de textes,* Doctoral thesis, Université Paris 7, 162 p. + vol. 2 (appendixes).

Laporte, Eric. 1990. Le dictionnaire phonémique DELAP. *Langue Française* 87, pp. 59-70, Paris: Larousse.

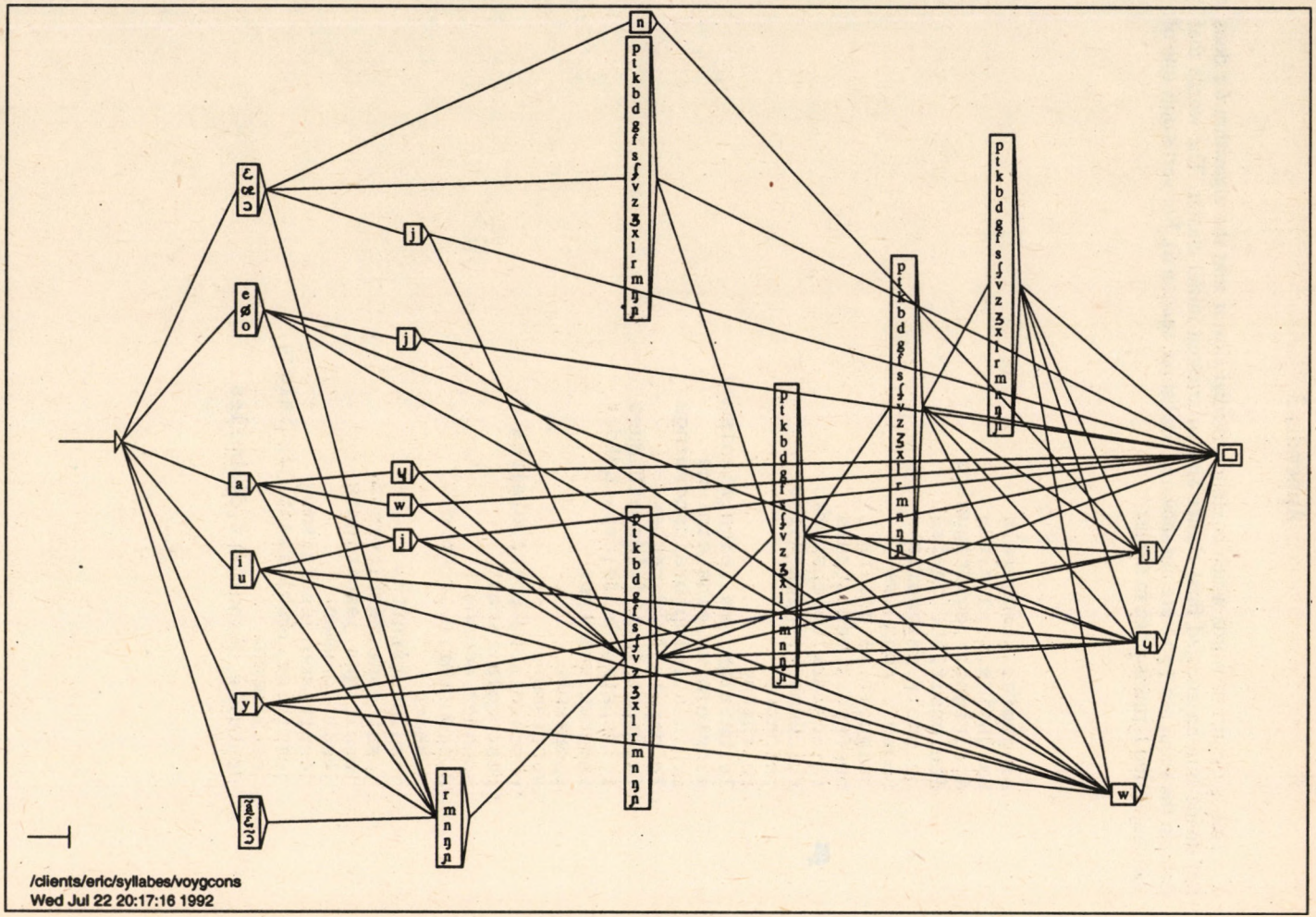
Liang, Franklin Mark. 1983. *Word Hy-phen-a-tion by Com-pu-ter,* Doctoral thesis, Stanford University, 85 p.

Mañas, José. 1987. "Word Division in Spanish", *Communications of the ACM,* vol. 30, no. 7, pp.612-616.

Appendix 1

When a consonant cluster begins with `[l]` or `[r]` and contains at least two other full consonants, then it contains only two other full consonants and the second one is `[l]` or `[r]`, like in `[filtr]` *filtre*. This is the beginning of the list of the exceptions to this rule.

```
absorption,absorptions
absorptivité,absorptivités
adsorption,adsorptions
antarctique,antarctiques
arctique,arctiques
austromarxisme,austromarxismes
calcschiste,calcschistes
circumantarctique,circumantarctiques
coarctation,coarctations
démarxiser
désorption,désorptions
dolce
dolcissimo
feldspath,feldspaths
feldspathique,feldspathiques
feldspathoïde,feldspathoïdes
feldwebel,feldwebels
hallstattien,hallstattiens
hallstattienne,hallstattiennes
hertz
hertzien,hertziens
hertzienne,hertziennes
holantarctique,holantarctiques
holarctique,holarctiques
hornblende,hornblendes
horst,horsts
hypermnésie,hypermnésies
hypermnésique,hypermnésiques
hypersplénisme,hypersplénismes
```


n
p t k b d g f s ʃ v z ʒ x l r m n ŋ ɲ
ɛ œ ɔ
e ø o
a
i u
y
ə œ̃ ɔ̃
j
ɥ
w
l r m n ŋ ɲ
□
/clients/eric/syllabes/voygcons
Wed Jul 22 20:17:16 1992

Appendix 3

This is the list of words which contain a medial cluster that the algorithm **6.c** does not divide into an attested final cluster and an attested initial cluster. The words that contain the sequence `[gj]` are not shown in the list (cf. section 6). For verbs only one of the conjugated forms is given in the list.

singspiel,singspiels
baignoire,baignoires
égratignoir,égratignoirs
éteignoir,éteignoirs
peignoir,peignoirs
rognoir,rognoirs
saignoir,saignoirs
beefsteak,beefsteaks
reichsmark,reichsmarks
reichsmark,reichsmarks
reichstag
reichstag
leibnizianisme,leibnizianismes
einsteinium,einsteiniums
brainstorming,brainstormings
feldspath,feldspaths
feldspathique,feldspathiques
feldspathoïde,feldspathoïdes
zemstvo,zemstvos
sportsmen
sportsman
postscolaire,postscolaires
postscolairement
sounder,sounders
discounter,discounters
makhzen
jésuite,jésuites
jésuitesse,jésuitesses
jésuitique,jésuitiques
jésuitiquement
jésuitisme,jésuitismes
jésuitise,jésuitisent,jésuitises
jésuitement
déshuile,déshuilent,déshuiles

Bilingual Reference Corpora: Creation, Querying, Applications

ELISABETTA MARINAI — CAROL PETERS — EUGENIO PICCHI

ABSTRACT

The paper discusses the importance of bilingual reference corpora as valid sources of real-world renderings of texts written in one language (L1) in a second (L2), and illustrates their potential for exploitation in various kinds of cross-language studies. A system that has been developed for the creation, management and interrogation of such corpora is presented, and the integration of this system in a Workstation providing facilities to query and extract information from both mono- and bilingual text archives and lexical databases is described.

1. INTRODUCTION

In the last decade there has been a considerable growth of interest in corpus construction as language reference corpora have become widely recognised as being important sources of information, not only for lexicographical purposes but also in many other types of linguistic studies including the acquisition of knowledge for natural language processing systems[1]. This activity has been encouraged by the recent technological evolution in data storage devices and optical character readers which means that the rapid acquisition, maintenance and management of large volumes of data at relatively low costs is becoming increasingly feasible. So far, efforts have been mainly concentrated on the construction and study of monolingual corpora, however, attention is now also being given to the creation of bilingual text archives. This interest is motivated by a growing awareness of the value of bilingual corpora as in-depth sources of documented evidence of how texts written on one language can be rendered in another, according to a number of contextual factors, such as style, register, domain, etc., and therefore of their potential for exploitation in many types of cross-language comparisons and investigations.

In the first place, such corpora can be considered as important tools in bilingual dictionary compilation, translating, and language learning activities. For

---

[1] See, for example, the success of important corpus-based dictionary projects such as COBUILD or the Trésor de la Langue Française and the recent decision to set up a Network of European Corpora (NERC) under the auspices of the European Community.

example, the bilingual lexicographer can use them in much the same way as the monolingual lexicographer refers to a monolingual language corpus: as a test-bed for his intuitions; to provide reliable material for examples. While a monolingual dictionary must provide an adequate representation of the lexical system of a given language, in the bilingual the focus is on representing the complex network of relationships between the lexical system of two languages. The necessity to represent a given headword on the basis of the different ways it can be rendered in the target language and also to specify the factors which affect the choice of the TL equivalent will influence the formulation of the SL entry and, in particular, can modify its breakdown into senses when compared to the equivalent monolingual entry. A bilingual reference corpus may well supply evidence to support the adjustment of a first SL analysis of an entry to meet the demands of the TL and can even provide a starting point for the formulation of new hypotheses on the relationships between the two languages in question.

Similarly, a bilingual corpus not only represents an important fount of inspiration for the translator, permitting him to find translations for words or expressions which are not listed in any dictionary, but could also provide important data for studies on the translation process. In fact, bilingual corpora should be considered as important repositories of data which make it possible to study many of the processes involved in transferring information, ideas and concepts from one language to another. Methods could be developed to evaluate data extracted from them at different levels: for example to examine changes in style and register or to investigate differences in sentence structure and discourse organization between a target language version to its source text. Such studies, useful in themselves in helping us to understand and improve the work of the translator, could also permit the acquisition of data that reflects important distinctions between the two languages in question and could lead to the formulation of generalizations on the relationships between them.

Data derived from analyses on bilingual corpora should also provide valuable input for MT systems. For example, Nagao (forthcoming) stresses the importance of including detailed collocational information in the transfer dictionaries of such systems: there are many specific expressions which must be translated in a specific way in a given TL and knowledge of this sort improves the quality of an MT system greatly. To acquire it many collocational expressions with their translations must be accumulated and bilingual texts are important sources of such data.

However, the construction of a large-scale bilingual text corpus is not a task to be undertaken lightly as it implies a considerable investment of time and resources. The goal must be a high quality corpus, sufficiently representative of the object it aims at modelling (whether the "entire" lexicon or particular sub-sets of it) and sufficiently large to provide valid data for a wide range of linguistic studies[2]. Before any decisions are taken, the criteria to be adopted when assembling corpus material must be carefully evaluated. Unfortunately, no hard and fast guidelines

---

[2] Although corpus representativeness is a clear goal for the future, important results can still be obtained in the meantime from material which is not balanced, as long as the user is aware of the type of data he is analysing. For instance, the availability of texts such as the Hansards transcripts of the Canadian parliamentary debates, has permitted much experimentation on the statistical processing of bilingual texts which would have been impossible without the availability of such enormous volumes of data (see, for example, work by Church and Gale, 1991)

are available which can be used to define the "correct" design criteria. Points to be considered include the definition of the sampling units, the text acquisition methods to be adopted, the range of language varieties to be sampled and the proportions of each to be included (depending on the level of representativeness to be guaranteed), the time period to be covered, the labelling of samples as source or target texts (and for target texts, some indication of translation status so that user can judge the value of the material he is handling), problems of copyright, and so on. Consultations with language specialists and potential users of the corpus are essential during this stage in order to evaluate the correctness of the approach adopted.

At Pisa, we have assembled a sample set of bilingual texts, selected to cover a number of different language varieties, ranging from scientific texts to poetry, from university text books to magazine articles. This set of texts was collected in the first place in order to provide a test-bed for a system which we have developed for the acquisition and management of bilingual corpora but should also provide useful data to assist us in the definition of valid design criteria which can then be used in a subsequent extension of these archives.

2. A System for Bilingual Corpus Management

A preliminary version of the system which we have developed for the automatic construction and retrieval of parallel contexts from bilingual text archives is described in (Marinai et al., 1991). The system has been designed to be run on sets of paired texts - where one is the translation equivalent of the other. The user queries either of the two sets of texts and, for any form or cooccurrences of forms which can be found in the set of texts for one language, retrieves parallel contrastive contexts from the other. At the moment, the languages treated by the system are Italian and English. However, the procedures have been designed to be generalizable; given the necessary lexical components they could be transported to run on other languages.

The system operates in two distinct steps. In the first stage, sets of bilingual texts are "synchronized" using morphological procedures and a bilingual electronic dictionary (derived from the Collins Concise English-Italian dictionary) for the two languages being treated in order to establish direct links between translation equivalents. Full details on how this text "synchronization" procedure operates can be found in Marinai et al.(*op.cit.*). These links are then stored with the texts in the bilingual archives to be used by the query system in the on-line construction of parallel contexts. For each L1 word or combination of words searched by the user, the parallel contexts for L2 are constructed in real time and displayed on the screen. The word(s) for which the contexts are being created are highlighted and other words that have been linked in the paired contexts can be optionally evidenced in a different colour. When there is no directly linked L2 form for the L1 word being searched, the two linked forms which are closest to the point calculated as the middle of the L2 context are evidenced in a different colour, as indicators of the likely position of the translation equivalent. The user can either search for single word forms or, using the morphological generator, for all the forms of a given lemma. The archives are considered to be symmetric; either of the two languages can be selected as L1. Bilingual concordances of interest can be printed out or saved in a separate file for future reference.

The system was first tested on the sample set of texts mentioned above, chosen to represent various types and styles of translation, in order to evaluate its

performance. The feed-back from the first results has enabled us to made certain improvements to the preliminary version of the system both in the definition of the zone in the L2 text which must be searched to find any translation equivalent of the L1 form being processed and in the procedure which creates the parallel contexts when the corpus is queried. "Wrong" links between falsely recognized translation equivalents which disturb context calculation are now identified and eliminated; the query procedure recalculates the parallel contexts on the basis of those links recognised as valid. We are now beginning to use the system to acquire larger text samples and even entire books in original and translated versions.

Figure 1 give an example of the results obtained for a query in which the Italian collocation "aria distratta" was searched in one of our bilingual sets of texts. In the figure, the words being searched and the words linked directly with them in the parallel contexts are shown in bold, whereas the indicators of the position of the translation equivalents appear in grey.

```
B.B.T. (Picchi)                          Synchro: Joyce - The Dubliners  V
{I}ARIA & {I}DISTRATTA
 1 {I} in mezzo alla strada c' era un uomo che suonava l' arpa dinanzi a
un cerchio di persone. Pizzicava le corde con aria distratta, lanciando di
tanto in tanto rapide occhiate ai nuovi venuti e poi levando gli occhi al
cielo, sempre con aria  I-Dublin6.172
   {E} the club a harpist stood in the roadway, playing to a little
ring of listeners. He plucked at the wires heedlessly, glancing quickly from
time to time at the face of each newcomer and from time to time, wearily
also, at the  E-Dublin6.190
 2 {I} da O'Neill era arrivata Miss Delacour. Si ricacciò in saccoccia
il berretto e entrò di nuovo in ufficio assumendo un' aria distratta. "Vi ha
cercato Mr Alleyne" disse in tono brusco il capufficio. "Ma dov' eravate?" L'
I-Dublin9.108
   {E} while he was out in O'Neill's. He crammed his cap back again
into his pocket and re_entered the office, assuming an air of
absent_mindedness. "Mr Alleyne has been calling for you", said the chief
clerk severely. "Where were you E-Dublin9.110
 3 {I} larga da quel Browne, perché in fondo non sarebbe cattivo."
Tremava per l' agitazione, adesso. Perché aveva quell' aria così distratta?
Non sapeva nemmeno come cominciare. Forse era anche lei tormentata da
qualcosa? Se soltanto si fosse rivolta  I-Dublin15.1535
   {E} because he' s not a bad fellow at heart." He was trembling now
with annoyance. Why did she seem so abstracted? He did not know how he could
begin. Was she annoyed, too about something? If she would E-Dublin15.1689
Enter Context No.>                                          F1 for help
```

Figure 1 Example of results of a search for all occurrences of "aria distratta" in the Bilingual Text Archives

In the next version of the system, to be released shortly, we intend to insert a function that will provide a preliminary statistical analysis of the results. For any SL word queried, the TL equivalents found in the parallel texts will be listed in descending order of occurrence for those words for which direct links have been made, followed by the number of parallel contexts in which no direct link has been found for the searched word in the TL text. Therefore, for the occurrences of **time** in our corpus the results will appear as shown in Figure 2, and the user can then select the particular contexts he wishes to view without necessarily having to scan through them all.

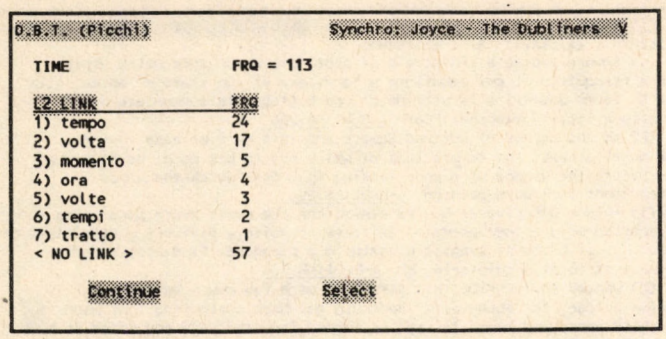


Figure 2 Frequencies for results of a search for parallel contexts of "time" in the Bilingual Text Archives

A bilingual corpus retrieval system of this type can be used to find much information on translation equivalents which is not given in bilingual dictionaries. One of the most important applications, both for the general user and for the lexicographer, is the extraction of information on the translation of bound or semi-bound expression, idioms, and frequent collocations.

We have already shown in Figure 1 how the general user can query the bilingual text system to find parallel contexts for cooccurrences of words in the set of texts of either language. A printed dictionary cannot contain an extensive range of such information, owing to space factors, and the lexicographer also has the problem of deciding which information should be included, or given priority. He can use the bilingual corpus system to search relevant data by first extracting right and left sorted concordances from the set of texts for just one language, considered as a monolingual corpus[3], in order to identify the most frequent collocates or set expressions for any given lemma. The relevant parallel contexts are then searched in order to access the appropriate translation equivalents. The bilingual lexicographer can then make an informed decision as to what should be included in his entry on the basis of the facts shown by the corpus. For example, a concordance run on our Italian set of texts in the bilingual corpus for **animo** reveals **stato/i d'animo** as one of the most frequent collocations and this expression becomes a strong candidate for inclusion in a dictionary entry.

The parallel contexts for the cooccurrence of **stato/i** and **animo** are shown in Figure 3. The lexicographer creating an entry for **animo** can "cut" the examples that interest him from the bilingual texts, edit them, and then "paste" them into his entry using the functions provided by the bilingual dictionary editor or update system, which is one of the components of the Bilingual Workstation (see below, Figure 4).

---

[3] In this kind of application, monolingual concordancing must be performed only on SL texts as translated texts can never give an entirely true representation of a language; they will always to some extent be dependent on or influenced by their source.

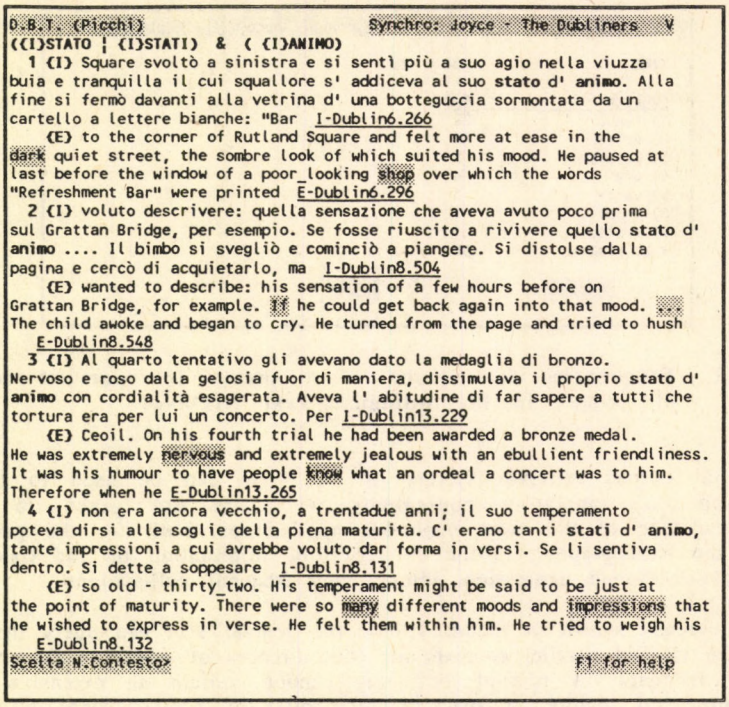


Figure 3 Cooccurrences of "stato/i" and "animo" in the Bilingual Text Archives

The bilingual text retrieval system described has been implemented for interactive consultation, mainly to meet the needs of lexicographers, translators or language learners. However, we are now examining methods by which the results can be synthesized so that the most probable translation equivalents for a given SL word or expression can be identified in its TL context and then extracted automatically.

3. THE BILINGUAL WORKSTATION

We are now working on the implementation of an integrated Bilingual Workstation which will provide the user with functions that permit him not just to query the bilingual text archives as described above but also to have on-line access to monolingual archives and corpora and to mono- and bilingual lexical databases for in-depth search operations. Our aim is to provide the user with fast, flexible, easy-to-use tools which permit him to interrogate and extract information from both the text and the dictionary components, navigating easily between them. Links between the text databases and the LDBs will permit lookup

on detailed morphological, syntactic and semantic information for any lexical item found in the texts; in addition semantic information, such as taxonomic data, derived from dictionary definition parsing operations can also be invoked to search the corpus data (see Calzolari and Picchi, 1986, for examples of this kind of text querying).

All the components included in the Workstation form part of the Pi-system, an open-ended modular set of tools, which has been developed to meet the various requirements of literary and linguistic text processing and analysis. The Workstation revolves around a core component constituted by the DBT, a textual database management and query system, which is implemented in different configurations to perform specific mono- and bilingual text and dictionary processing activities (see Picchi, 1991). Other components are: the MLDB, a multilingual integrated lexical database system first described in Marinai et al. (1990) - the lexical components of the MLDB include the Italian Machine Dictionary (mainly based on the Zingarelli Italian Dictionary), the Garzanti 'Nuovo Dizionario Italiano', and the Collins Concise Italian/English, English/Italian Dictionary (it is hoped to add an English LDB shortly); the Bilingual Corpus Management System described in Section 3 above; and English/Italian Bilingual and Italian Monolingual Text Archives (for information on the Italian Reference Corpora now under construction at the ILC-CNR, Pisa, see Bindi et al. (1991)). The configuration of the Workstation is shown in the following figure.

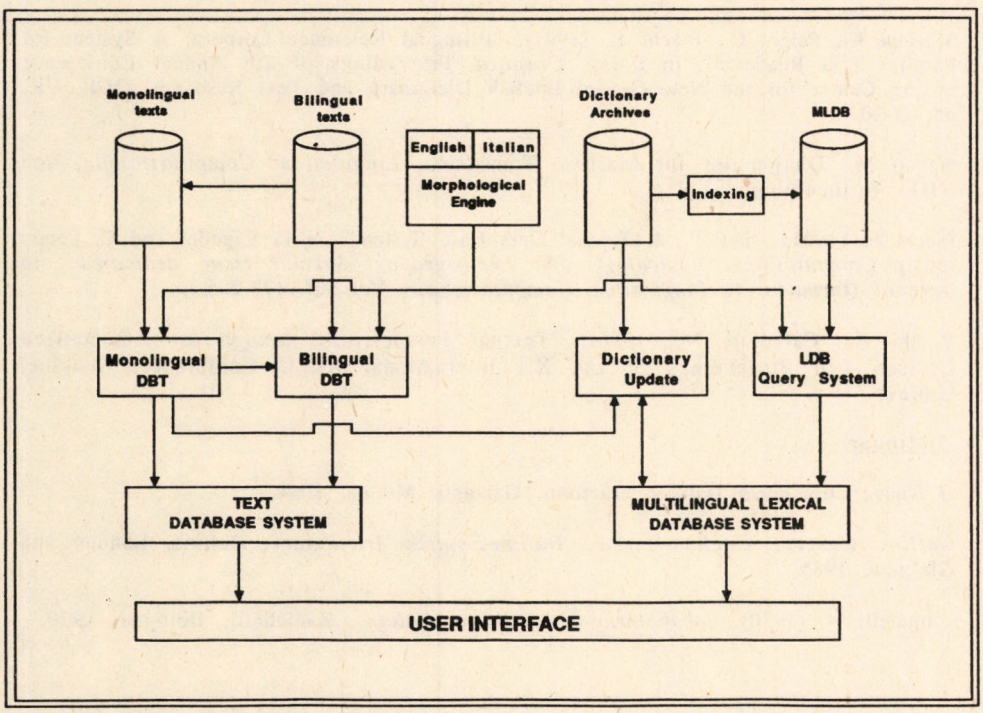


Figure 4 The Bilingual Workstation

This Workstation can be employed in many types of activities: by the lexicographer, by the translator, by the language learner, or indeed by any user wanting to exploit to the full the possibility of being able to dynamically access, browse and extract the different kinds of linguistic information contained in our dictionary and text databases. The user can move freely and rapidly from one component to another, using the information extracted from one to query or supplement information contained in another. The entire system is menu-driven; the user is guided in his use of each component by a standardized set of command menus and context sensitive Helps can be invoked to explain the functionality of each command.

REFERENCES

Bindi R., Monachini M., Orsolini P. (1991), "Italian Reference Corpus: General Information", Technical Report, ILC-CNR-TLN.

Church, K, Gale, W. (1991): "Concordances for Parallel Text", in *Using Corpora.*, Proceedings of the 7th Annual Conference of the Centre for the New Oxford English Dictionary and Text Research, OUP, UK, 42-62.

Marinai E., Peters C., Picchi E. (1990), "The Pisa Multilingual Lexical Database System", Esprit Basic Research Action No. 3030, Twelve Month Deliverable, ILC-ACQ-2-90, Pisa, 61p.

Marinai E., Peters C., Picchi E. (1991), "Bilingual Reference Corpora: A System for Parallel Text Retrieval", in *Using Corpora*, Proceedings of 7th Annual Conference of the Centre for the New Oxford English Dictionary and Text Research, OUP, UK, pp 63-70.

Nagao M., Dictionaries for Machine Translation, *Linguistica Computazionale*, Vol VIII, forthcoming.

Picchi E. (1991). "D.B.T.: A Textual Data Base System", in L. Cignoni and C. Peters (eds.), *Computational Lexicology and Lexicography. Special Issue dedicated to Bernard Quemada. I., Linguistica Computazionale*, Vol VII, 177-205.

Picchi E., Calzolari N. (1986), "Textual Perspectives through an Automatized Lexicon", in Proceedings of the XII International ALLC Conference, Slatkine, Geneva.

Dictionaries

Il Nuovo Dizionario Italiano Garzanti, Garzanti, Milano, 1984.

Collins Concise English-Italian, Italian-English Dictionary, Collins, London and Glasgow, 1985.

Zingarelli N. (1970), *Vocabolario della Lingua Italiana*, Zanichelli, Bologna, 1970.

Tagged Corpora: A Query System

MONICA MONACHINI - EUGENIO PICCHI

Abstract

The work illustrated in this paper is being carried out within the EC project "Network of European Textual Reference Corpora" (NERC). A computational tool which has been designed to provide corpus users with an interactive aid to retrieve information contained in an annotated textual data-base is presented and the structure of the operating system and the data query procedures are described in detail. The tool is a component of a more vast system, the Pi-system, a set of tools developed for literary and linguistic text-processing purposes at the Institute for Computational Linguistics of the Italian National Research Council, Pisa.

Keywords

Linguistic Resources, Corpus Linguistics, Annotated Corpora, Computational Tools

1. Introduction

The work presented in this paper has been inspired by two important trends which have emerged in the last few years: the considerable attention being paid by the scientific community to large, shareable and interchangeable linguistic resources and the wide-spread increasing interest in large text corpora (Calzolari & Zampolli, 1990). Text corpora are requested by a growing number of academic and industrial users for many kinds of purposes, either scientific or commercial (Monachini, 1992). They are useful, to mention a few examples, in NLP activities, in psycho-linguistics, in lexicography and lexicology and in language engineering. Corpora can integrate the lack of certain types of information in printed, Machine Readable Dictionaries, and in Computational Lexica (Bindi et al., 1992).

The collection of enormous quantities of linguistic data and in particular of text corpora of million of words has software implications (Sinclair, 1991). The creation of tools for data analysis and classification, and the storage of the results of these operations are economically relevant tasks.

There is thus a need for systems which, on the one hand offer rapid, direct access to all the different elements contained in a text, and also to the results of analyses performed on the text and, on the other, tools which permit the information to be reused[1] by applications for knowledge extraction from corpora (Bindi et al. 1991).

The more powerful and interactively oriented the tool, the better.

At our Institute, procedures which operate on textual and lexical data have been studied extensively over the last years. The result has been the Pi-system, a set of tools developed to meet the needs of a wide range of literary and linguistic text processing activities. The kernel of this system is the textual database management system, DBT (Picchi, 1983, or 1991).

The system presented in this paper is a component of the above Pi-system and has been designed to permit the user to access and consult text corpora interactively, according to the interests of the user. Particular procedures have been implemented so that many different types of linguistic information stored explicitly and implicitly in the texts can be searched and retrieved.

In this way, vast quantities of valuable linguistic information, otherwise inaccessible, can be consulted.

The work presented here is being carried out within the project "Network of European Reference Corpora" (NERC) whose the main goal is to evaluate the feasibility of common criteria for composition, standards for coding, strategies, methodologies and tools as far as corpora are concerned (Zampolli, 1990).

---

[1] The notion of "reusability" has become central in research on large linguistic resources (see Calzolari & Zampolli, 1990). In the collection of million of words it is important to know how to retrieve information and what to do with it if we do.

2. A Database for Tagged Corpora

2.1 Tags, Tagging and Tagged Corpora

Among the possible annotation practices, i.e. the practice of adding interpretative linguistic information to a corpus at various levels by some kind of coding (Leech, 1992), grammatical tagging is the most known and familiar because it can be performed largely automatically (Garside et al., 1987; Marcus & Santorini, 1992). Tagging classifies morphosyntactically each word-form in a text, labelling it with a tag, i.e. a simple or complex code which encodes information on the part of speech (PoS) and the morphological features.

Tagged corpora are useful, in general, in NLP, where the knowledge derived from tagged corpora is used to improve rule-based parsers or to provide a basis for parser with a statistical approach.

They play a crucial role in studies on the lexicon, where word class and meaning are often dependent one on the other (Sinclair, 1991 and 1992).

However, tagged corpora must not only be considered as a step towards parsing, they are important in themselves, as repositories of linguistic phenomena, for example, real sequences of grammatical categories. As such, they are useful in supplying documented evidence which can be used to back up theoretical suppositions on the probability of particular word order.

2.2 Handling Tagged Corpora

The tool we are presenting in this paper has been conceived as a computational aid for corpus users enabling them to handle tagged corpora accessing interactively the entire text, the single words and all the various levels of linguistic annotation: the tag attached to each word form, the word+tag pair, and, when the text is lemmatised, any lemma linked to a word form or the lemma+form set.

The query system is menu-driven and user-friendly. The user can easily formulate simple and complex queries without prior knowledge of the command language: during a query session, the screen presents a menu of all the commands which can be used; commands are entered by selecting them from the menu or using their key letters which are highlighted. In response to a query, the system displays the context(s) in which the object of the query is instantiated.

2.3 Structure of the Database

The tagged corpus data to be acquired by the system are structured in DBT format: this implies an analysis of the text to recognize the different elements composing it and its transformation into DBT structure. For a full description of procedures for the DBT acquisition and structuring of texts (Picchi, 1991).

The system operates on textual data at various levels:

- the *text*, seen as coherent string/span of written natural language .
- the *tag*, i.e. a simple or complex code which encodes information on the part of speech (PoS) and the morphological features, associated with each word form
- the *lemma*, which connects all instances of word forms of the same lemma (the lemma can be seen either as equivalent of the dictionary headword without solving the polysemy, or at the word sense level where polysemy has been disambiguated)

The data is stored in archives structured as follows:

Base level: the archive contains a structured copy of the text in DBT format.

If the text is tagged, each word in the text is stored together with its tag.

Index level at this level, we have indices for all the word forms in the text together with their tags. The text can be viewed and queried with or without the tags. The possibility of reverting to the raw corpus if necessary, is considered very important (Leech, 1992).

Lemma Index Level this level is activated when working on lemmatized texts. It contains all the lemmas for the word forms in the text and explicit links to the forms, and to the related occurrences. The lemmas are only stored in the system in the lemma index and not explicitly at the text archive level.

The system operates automatically activating the "word path" or the "word+tag path", according to the data or operational mode selected by the user.

2.4 *Query system*

The query system is based on a tool box of corpus access functions; software modules from the tool box can be integrated in procedures and applications without any need to enter into the core of the function. The engine of the tool box is the query system which allows the whole corpus to be interrogated on-line, with specialized search functions for each level of query: 1) Base Functions, 2) Tag Functions, 3) Lemma Functions.

2.4.1 Base Functions

These functions operate at the unannotated text level:

Search: the following types of look-up can be made:

- simple search functions can be entered using strings of characters or wildcards. It is possible to search: i) words beginning, ending with or containing a given string; ii) words beginning, ending with, containing a given string or another given string.

- complex search functions can be used to define co-occurrences of two or more given items in the same context: such functions are known as 'word families' in the system: families are defined using w for item and f for family with the operators AND=&, OR=/, NOT=⌐. The user can retrieve contexts with: i) cooccurrences of word1 & word2, word1 OR word2, word1 but NOT word2, ii) word3 with either word1 OR word2, iii) cooccurrences of family1 together with word4 and word5, and so on.

View: contexts which are obtained as results of a query can be displayed sorted in
(i) text order
(ii) left order
(iii) right order according to the keyword
The size of context can be modified by the user, as he wishes.

2.4.2 Tag Functions

These functions switch on the word+tag path: the system operates on the pair word+tag, considered as a simple element. The same search and view facilities as described above for the base functions are available.

In addition, searches can be made which distinguish between homographic forms on the basis of their tag. The set of tags on which the query is to operate can be defined interactively by the user (see Figure 2).

The context(s) retrieved as the result of a query can be displayed with or without the tags associated with each word form (see the Figure below).

Restrictions: in order to permit the querying of linguistic phenomena, which cannot be found by simple word+tag searches, additional functions have been implemented which permit conditions to be imposed on a query: these "restriction" rules are applied to the context(s) to be retrieved: only those contexts which satisfy the conditions imposed by the operator are accepted.

This facility makes it possible to operate selections in terms of grammatical structures, here conceived as sequences of tags. Particular sequences can be defined in order to create "macros" which can be used within the Restriction rules: the following sequence, DT [JJ*] NN [JJ*], constitutes a possibile macro which could be used to search a Noun Phrase (NP).

2.4.3 Lemma Functions

As we are convinced that, especially when working with highly inflected languages it

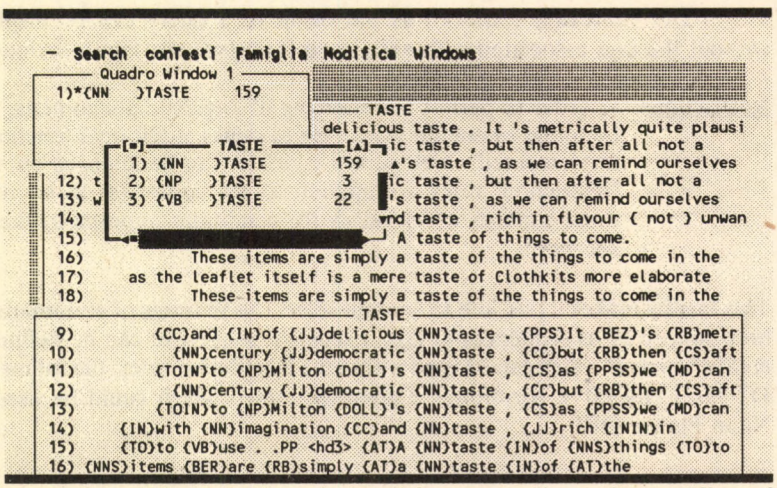
can often (even if not always: see, e.g. Bindi et al., 1991) be useful to operate on lemmas rather than on word forms, in order to limit the dispersion of information in inflected forms.

A set of functions working at the lemma level has thus been implemented: these functions can be used to make searches on the form-lemma set, starting either from the form or from the lemma. The same types of search and view functions as described above can be used although the restriction rules can only be applied at the word-tag level, not at the lemma level, as the lemmas are only stored in the lemma index and not in the text archive.

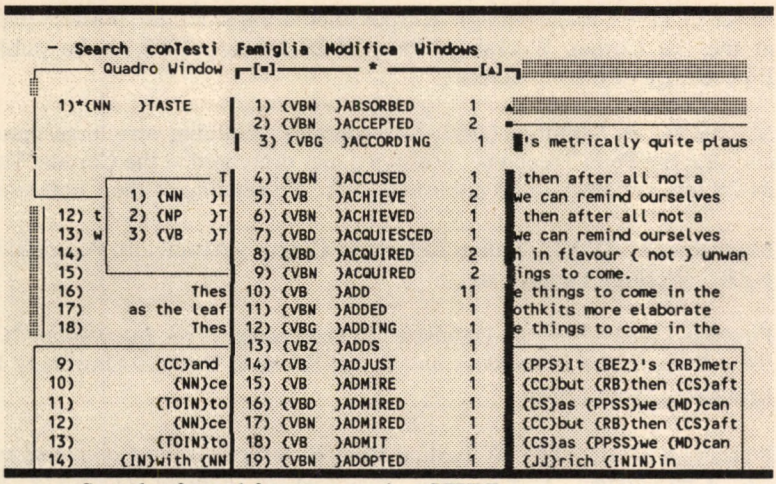
3. Concluding remarks

In the present paper we have described a system designed to meet the needs of corpus users: it provides functions for interactive browsing and navigating through raw and/or annotated data at all the different levels of linguistic analysis.
We feel that a system of this type has an immediate application in computational lexicography and lexicology; it should also be particularly useful in the storage, acquisition and restructuring of new kinds of analyses on textual data, such as syntactic and semantic analyses.

Figures



1 Interactive selection of a word form according to its tag. View of contexts with or without tags.



2 Search of word-forms tagged as VERB

References

Bindi R., N. Calzolari, M. Monachini, V. Pirrelli (1991): "Lexical Knowledge Acquisition from Textual Corpora: A Multivariate Statistic Approach as an Integation to Traditional Methodologies", in *Using Corpora Proceedings of the Seventh Annual Conference of the UW Centre for the NEW OED and Text Research,* OUP, Oxford, 170-196.

Bindi R., N. Calzolari, M. Monachini, V. Pirrelli, A. Zampolli (1992, forthcoming): "Corpora and Computationa Lexica: Integration of Different Methodologies of Lexical Knowledge Acquisition, position paper presented at the Corpus Workshop, Pisa, 24-26 January 1992, to be published in the Proceeding of the Conference.

Calzolari N., A. Zampolli (1990): "Lexical DataBases and Textual Corpora a trend of Convergence between Computational Linguistics and Literary and Linguistic Computing", in *Proceedings of ALLC-ACH Conference*, S. Hockey and N. Ide eds., OUP, Oxford.

Garside R., G.Leech, G.Sampson (1987): *The Computational Analysis of English - a corpus based approach*, Longman, London.

Leech G. (1992, forthcoming): "Corpus annotation schemes", position paper presented at the the Corpus Workshop, Pisa, 24-26 January 1992, to be published in the Proceedings of the Conference.

Marcus M., B. Santorini (1992, forthcoming); "Building very large natural language corpora: the Penn Treebank", position paper presented at the Corpus Workshop, Pisa, 24-26 January 1992, to be published in the Proceeding of the Conference.

Monachini M. (1992); "Italian Reference Corpus - Actual and Potential User Needs", NERC Working Paper, Pisa.

Picchi E. (1983): "Textual Database", in *Proceedings of the International Conference on Data Bases in the Humanities and Social Sciences,* Rutgers University Library, New Brunswick, New Jersey.

Picchi E. (1991): "DBT: A Textual Data Base System", in *Computational Lexicology and Lexicography. Special issue dedicated to Bernard Quemada. i, Linguistica Computazionale, vii*, L. Cignoni and C. Peters eds., Pisa.

Sinclair J.M. (1991): "The automatic Analysis of Corpora", for the Nobel Symposium on Corpus Linguistic, Birmingham.

Sinclair J.M. (1992, forthcoming): "Lexicographer's Needs", position paper presented at the Corpus Workshop, Pisa, 24-26 January 1992 and to be published in the Proceedings of the Conference.

Zampolli A. (1990): "Project Definition for the Constitution of a Network of European Textual Reference Corpora, ILC-CNR, Pisa.

Formalisation des données lexico-syntaxiques dans le dictionnaire

NAM JEE-SUN

résumé

Un dictionnaire électronique doit contenir des ensembles de données morphologiques et syntaxiques caractérisant tous les items répertoriés. Dans un lexique-grammaire, ces items sont classés en fonctions des constructions fondamentales dans lesquelles ils entrent.

Nous avons étudié un ensemble d'adjectifs coréens (prédicatifs) prenant un complément essentiel en *-wa*. Dans cet ensemble, nous avons dégagé, par des critères formels comme l'inversion des actants ou les alternances de postpositions, une classe d'adjectifs qu'on peut définir comme "symétriques". Nous étudions ici les problèmes rencontrés pour isoler une telle classe et tentons de préciser en quoi ces adjectifs symétriques se distinguent des autres.

0. PROBLEMATIQUE

La construction d'un dictionnaire (électronique ou non) contenant un ensemble structuré d'informations lexico-syntaxiques est une étape obligée dans le développement du traitement des langues assisté par ordinateur. L'enregistrement des données d'une langue naturelle doit donc être effectué d'une manière systématique et accessible à l'informatique, ce qui demande certains principes cohérents dans la détermination des entrées lexicales dans le dictionnaire et dans la représentation syntaxique des phrases où une entrée donnée peut apparaître. L'état actuel des dictionnaires du coréen n'est pas adéquat pour une telle informatisation, tant du point de vue lexical que syntaxique : par exemple, la catégorie grammaticale "*Adjectif*" est répertoriée selon une intuition sémantique, et en conséquence les frontières de l'adjectif ne se superposent pas dans des dictionnaires. En particulier, se pose le problème de la séparation de l'adjectif par rapport au verbe, puisque l'adjectif, comme le verbe, sert de prédicat, seul, sans prendre la copule *ita* [être]. Par contre, le substantif doit être accompagné d'une copule pour fonctionner comme attribut. Soit :

(1) 민우는 (공부하-ㄴ다 + 착하-다)
*Minu-nîn (kongpuha-***nta** + *chakha-***ta***)*
Minu-nominatif (travailler-suffixe terminal + gentil-suffixe terminal)
(Minu (travaille + est gentil))

(2) 민우는 변호사-이-다
*Minu-nîn pyônhosa-***i-ta**
Minu-nominatif avocat-**ita[être]**-suffixe terminal
(Minu est avocat)

Il nous faut donc des critères formels pour délimiter l'ensemble appelé *Adjectif* de façon reproductible. Nous avons utilisé comme critère les **suffixes terminaux** (du mode déclaratif) qui s'ajoutent systématiquement à tous les éléments prédicatifs : les **verbes** prennent *-nta* comme suffixe terminal (*St*), les **adjectifs** interdisent cette forme de suffixe (par exemple, pour (1) on observe : *Minu-nîn chakha-(ta* + ***nta)* [Minu est gentil]). La grande majorité des entrées classées *Adjectif* dans les dictionnaires actuels présente cette caractéristique, ce qui nous a permis d'établir un corpus destiné à l'étude syntaxique des constructions adjectivales simples.

La structure des phrases adjectivales est définie par les compléments propres à chaque adjectif et plus précisément par les **postpositions appropriées** à ces compléments. Notre étude se limitera à une classe d'adjectifs qui prennent le complément en *-wa*. La construction sera donc de forme :

(3) N_0-*nmtf* N_1-**wa** *W Adj-St*

(où *nmtf* est une postposition du cas nominatif, **wa** est une des postpositions de complément, qui peut se traduire en *avec*, *à* ou *de* en français. **wa** se réalise en deux variantes ***wa*** et ***kwa*** selon l'absence ou la présence de la consonne finale du substantif accompagné, *W* correspond à des adverbiaux ou compléments éventuels et *St* est un suffixe terminal du mode déclaratif.)

1. STRUCTURE N_0-nmtf N_1-wa Adj-St

1.1. *N-wa* dans la construction symétrique

Le complément en *-wa* caractérise certains types de verbes et d'adjectifs. Dans les phrases suivantes :

(4a) A 선은 B 선과 평행하다
A sôn-în B sôn-kwa phyônghängha-ta
A ligne-nmtf B ligne-**wa** parallèle-St
(La ligne A est parallèle à la ligne B)

(4b) 민우는 인아와 이혼했다
Minu-nîn Ina-wa ihonha-ôs'ta
Minu-nmtf Ina-**wa** divorcer-Passé/St
(Minu a divorcé d'avec Ina)

le complément en *-wa*, obligatoire, est en rapport de **symétrie**[1] avec le sujet. Le rapport dit symétrique se justifie par le fait que (4a) et (4b) font respectivement une paire d'équivalence avec (5a) et (5b) :

(5a) B 선은 A 선과 평행하다
B sôn-în A sôn-kwa phyônghängha-ta
B ligne-nmtf A ligne-**wa** parallèle-St
(La ligne B est parallèle à la ligne A)

(5b) 인아는 민우와 이혼했다
Ina-nîn Minu-wa ihonha-ôs'ta
Ina-nmtf Minu-**wa** divorcer-Passé/St
(Ina a divorcé d'avec Minu)

L'un implique nécessairement l'autre : la relation entre les deux phrases peut être considérée comme une relation transformationnelle[2] :

(6) N_0-*nmtf* N_1-**wa** *W Adj-St*
= N_1-*nmtf* N_0-**wa** *W Adj-St*

Notons que cette opération ne s'applique qu'à la construction à complément en *-wa*. Les phrases ayant d'autres compléments comme :

(7a) 민우는 인아에게 지극하다
*Minu-nîn Ina-***eke** *cikîkha-ta*
Minu-nmtf Ina-**e** attentionné-St
(Minu est attentionné envers Ina)

(7b) 민우는 인아를 사랑한다
*Minu-nîn Ina-***lîl** *salangha-nta*
Minu-nmtf Ina-**Acc** aimer-St
(Minu aime Ina)

ne sont pas synonymes, respectivement, de :

---

[1] Les constructions à verbe symétrique ont été étudiées en français dans A. Borillo (1971), J.-P. Boons, A. Guillet et Ch. Leclère (1976) et en coréen dans Hong Chai-Song (1987).
[2] Suggestion de Maurice Gross.

(8a) 인아는 민우에게 지극하다
*Ina-nîn Minu-***eke** *cikîkha-ta*
Ina-nmtf Minu-**e** attentionné-St
(Ina est attentionnée envers Minu)

(8b) 인아는 민우를 사랑한다
*Ina-nîn Minu-***lîl** *salangha-nta*
Ina-nmtf Minu-**Acc** aimer-St
(Ina aime Minu)

Pourtant, la relation décrite en (6) n'est pas une condition suffisante pour définir la construction symétrique. La phrase suivante, par exemple :

(9) 민우는 인아와 일했다
Minu-nîn Ina-wa ilha-ôs'ta
Minu-nmtf Ina-**wa** travailler-Passé/St
(Minu a travaillé avec Ina)

qui est synonyme de :

(10) 인아는 민우와 일했다
Ina-nîn Minu-wa ilha-ôs'ta
Ina-nmtf Minu-**wa** travailler-Passé/St
(Ina a travaillé avec Minu)

n'implique pas un rapport symétrique entre N_0 et N_1. Le complément en *-wa* n'y est pas obligatoire, alors qu'il l'est souvent dans la construction symétrique. Ainsi on observe :

(11a) 민우는 (인아와 + E + 혼자서) 일했다
Minu-nîn (Ina-wa + E + honcasô) ilha-ôs'ta
Minu-nmtf (Ina-**wa** + E + seul) travailler-Passé/St
(Minu a travaillé (avec Ina + E + seul))

(11b) 민우는 (인아와 + *E + *혼자서) 비슷하다
*Minu-nîn (Ina-wa + *E + *honcasô) pisîsha-ta*
Minu-nmtf (Ina-**wa** + E + seul) ressemblant-St
(Minu est ressemblant (à Ina + *E + *seul))

Par ailleurs, si ce complément se trouve conjoint au sujet comme dans :

민우와 인아는 일했다
Minu-wa Ina-nîn ilha-ôs'ta
[Minu-**et** Ina]-nmtf travailler-Passé/St
(Minu et Ina ont travaillé)

la phrase est ambiguë : soit synonyme de (9) et (10) comme "ils ont travaillé ensemble", soit une forme réduite à partir des deux phrases "Minu a travaillé et Ina a travaillé" par l'opération de coordination. Dans ce dernier cas, il n'existe aucune interaction entre N_0 et N_1.

En fait, la construction symétrique est paraphrasable par une phrase à sujet coordonné comme :

민우와 인아는 비슷하다
Minu-wa Ina-nîn pisîsha-ta
[Minu-**et** Ina]-nmtf ressemblant-St
(Minu et Ina sont ressemblants)

mais elle n'est jamais une forme réduite des deux phrases :

*민우는 비슷하고 인아는 비슷하다
Minu-nîn pisîsha-ko Ina-nîn pisîsha-ta
Minu-nmtf ressemblant-Coord Ina-nmtf ressemblant-St
(*Minu est ressemblant et Ina est ressemblante)

Cependant, on n'observe pas de complément d'**accompagnement**[3] du type (9) dans la construction adjectivale, alors que l'on le trouve fréquemment dans la phrase verbale. Cela nous suggère que la relation transformationnelle des phrases (6) soit une condition suffisante, du moins dans la définition des constructions symétriques adjectivales. Regardons maintenant le cas suivant.

1.2. *N-wa* dans la construction pseudo-symétrique

Soit :

(12a) 민우는 인아와 의외로 냉정했다
Minu-nîn Ina-wa îïölo nängcôngha-ôs'ta
Minu-nmtf Ina-**wa** incroyablement froid-Passé/St
(Minu a été incroyablement froid avec Ina)

(12b) 진오는 민우와 무척 사무적이다
Gino-nîn Minu-wa muchôk samucôki-ta
Gino-nmtf Minu-**wa** très professionnel-St
(Gino est très professionnel avec Minu)

Les sujets n'en sont pas reliés aux compléments en *-wa* par un rapport de symétrie. La postposition *-wa* doit se traduire plutôt comme la postposition *-e* datif. Ainsi, on ne trouve pas d'équivalence entre (12a) (12b) et (13a) (13b) respectivement :

(13a) 인아는 민우와 의외로 냉정했다
Ina-nîn Minu-wa îïölo nängcôngha-ôs'ta
Ina-nmtf Minu-**wa** incroyablement froid-Passé/St
(Ina a été incroyablement froide avec Minu)

(13b) 민우는 진오와 무척 사무적이다
Minu-nîn Gino-wa muchôk samucôki-ta
Minu-nmtf Gino-**wa** très professionnel-St
(Minu est très professionnel avec Gino)

Les phrases (12a) et (12b) sont plutôt synonymes de :

(14a) 민우는 인아에게 의외로 냉정했다
Minu-nîn Ina-eke îïölo nängcôngha-ôs'ta
Minu-nmtf Ina-**e** incroyablement froid-Passé/St
(Minu a été incroyablement froid **envers** Ina)

---

[3] Pour ce type de complément, voir A. Borillo 1971 : ce complément est paraphrasable par *en compagnie de N* ; et Hong C.-S.1987 pour le coréen : cette construction accepte *hamk'e* (en compagnie / ensemble).

(14b) 진오는 민우에게 무척 사무적이다
Gino-nîn Minu-eke muchôk samucôki-ta
Gino-nmtf Minu-**e** très professionnel-St
(Gino est très professionnel **envers** Minu)

où le procès est clairement orienté.

Or, si on insère un élément lexical *sôlo* ("l'un Prép l'autre" ou "et réciproquement") dans (12), un rapport de **réciprocité** s'installe entre le sujet et le complément en *-wa* :

(15a) 민우는 인아와 의외로 서로 냉정했다
Minu-nîn Ina-wa îiölo **sôlo** *nängcôngha-ôs'ta*
Minu-nmtf Ina-**wa** incroyablement **sôlo** froid-Passé/St
(Minu a été incroyablement froid avec Ina, et réciproquement)

(15b) 진오는 민우와 서로 무척 사무적이다
Gino-nîn Minu-wa **sôlo** *muchôk samucôki-ta*
Gino-nmtf Minu-**wa** **sôlo** très professionnel-St
(Gino est très professionnel avec Minu, et réciproquement)

alors que l'insertion de *sôlo* dans la phrase à complément en *-e* datif (14) est interdite et que, par conséquent, on n'observe pas de symétrie entre des actants :

(16a) *민우는 인아에게 의외로 서로 냉정했다
Minu-nîn Ina-eke îiölo* **sôlo *nängcôngha-ôs'ta*
Minu-nmtf Ina-**e** incroyablement **sôlo** froid-Passé/St
(*Minu a été incroyablement froid **envers** Ina, et réciproquement)

(16b) *진오는 민우에게 서로 무척 사무적이다
Gino-nîn Minu-eke* **sôlo *muchôk samucôki-ta*
Gino-nmtf Minu-**e** **sôlo** très professionnel-St
(*Gino est très professionnel **envers** Minu, et réciproquement)

Dans (15), un rapport réciproque s'introduit donc par l'unité *sôlo* et non par l'adjectif même. Mais le complément en *-wa* n'y est plus remplaçable par le complément en *-e*. En revanche, *sôlo* est susceptible de prendre la postposition *-e*, qui est normalement réservée aux substantifs et aux pronoms :

(17a) 민우는 인아와 의외로 서로(E + 에게) 냉정했다
Minu-nîn Ina-wa îiölo sôlo(E + eke) nängcôngha-ôs'ta
Minu-nmtf Ina-**wa** incroyablement **sôlo(E + e)** froid-Passé/St
(Minu a été incroyablement froid avec Ina, et réciproquement)

(17b) 진오는 민우와 서로(E + 에게) 무척 사무적이다
Gino-nîn Minu-wa sôlo(E + eke) muchôk samucôki-ta
Gino-nmtf Minu-**wa** **sôlo(E + e)** très professionnel-St
(Gino est très professionnel avec Minu, et réciproquement)

Le fait que *sôlo* prenne cette postposition suggère que son statut grammatical est plutôt celui d'un pronom (réciproque) qui joue un rôle d'actant : (17a) et (17b) peuvent donc être considérées comme des formes dérivées des phrases coordonnées :

(18a) 민우는 인아에게 의외로 냉정했고, 인아도 민우에게 의외로 냉정했다
Minu-nîn Ina-eke îïölo nängcôngha-ôs'-ko Ina-to Minu-eke îïölo nängcôngha-ôs'ta
Minu-nmtf Ina-**e** incroyablement froid-Passé-Coord Ina-aussi Minu-**e** incroyablement froid-Passé/St
(Minu a été incroyablement froid **envers** Ina et Ina aussi a été incroyablement froide **envers** Minu)

(18b) 진오는 민우에게 무척 사무적이고, 민우도 진오에게 무척 사무적이다
Gino-nîn Minu-eke muchôk samucôki-ko Minu-to Gino-eke muchôk samucôki-ta
Gino-nmtf Minu-**e** très professionnel-Coord Minu-aussi Gino-**e** très professionnel-St
(Gino est très professionnel **envers** Minu et Minu aussi est très professionnel **envers** Gino)

Les adjectifs qui entrent dans cette construction, construction où *sôlo* est indispensable pour introduire la réciprocité entre deux actants, sont à distinguer des adjectifs dits *symétriques* dont la construction n'est pas une réduction de la phrase coordonnée. Nous appelons ce type de construction **construction pseudo-symétrique** : elle ne subit pas la transformation proposée au départ, si *sôlo* n'est pas présent. Les conditions qui caractérisent cette construction devront être étudiées plus en détail.

2. STRUCTURE N_0-nmtf N_1-wa N_2-i Adj-St

Revenons à la relation (6) qui définit la construction symétrique adjectivale :

*N_0-nmtf N_1-**wa** W Adj-St*
= *N_1-nmtf N_0-**wa** W Adj-St*

Le symbole *W* n'y représente pas seulement des adverbiaux, mais aussi d'autres compléments éventuels. Ainsi on a :

(19) 민우는 인아와 성격이 다르다
Minu-nîn Ina-wa **sôngkyôk-i** *talî-ta*
Minu-nmtf Ina-**wa** caractère-**i** différent-St
(Minu est différent de Ina **de caractère**)

Ces compléments sont toujours en *-i* dans la construction adjectivale. Par contre, dans la construction verbale, ce sont des accusatifs (en *-lîl*) comme :

(20a) 민우는 인아와 역할을 바꾸었다
Minu-nîn Ina-wa **yôkhal-îl** *pak'u-ôs'ta*
Minu-nmtf Ina-**wa** rôle-**Acc** changer-Passé/St
(Minu a changé **de rôle** avec Ina)

(20b) 민우는 먼저 할일을 나중 할일과 바꾸었다
Minu-nîn **môncô halil-îl** *nacung halil-kwa pak'u-ôs'ta*
Minu-nmtf premier tâche-**Acc** deuxième tâche-**wa** changer-Passé/St
(Minu a changé **la première tâche** avec la deuxième)[4]

---

[4] **Quand on permute ces deux compléments :**
(i) 민우는 나중 할일과 먼저 할일을 바꾸었다
Minu-nîn nacung halil-kwa môncô halil-îl pak'u-ôs'ta

Quand on a un troisième substantif, le rapport de symétrie s'établit d'une façon plus compliquée : soit entre le sujet et le complément en *-wa* comme dans (20a), soit entre les compléments (complément accusatif et complément en *-wa*) comme dans (20b). Dans la construction adjectivale aussi, le troisième substantif accompagné d'un *-i* concerne le rapport de symétrie, soit entre le sujet et le complément en *-wa*, soit entre les compléments (en *-i* et en *-wa*).

2.1. *N-i* est un élément restructuré

Voici des exemples de la structure N_0-*nmtf* N_1-*wa* N_2-*i Adj-St* :

(21a) 민우는 인아와 성격이 다르다
Minu-nîn Ina-wa **sôngkyôk-i** *talî-ta*
Minu-nmtf Ina-**wa** caractère-**i** différent-St
(Minu est différent de Ina **de caractère**)

(21b) 민우는 왼손이 오른손과 다르다
Minu-nîn **ŏn son-i** *olîn son-kwa talî-ta*
Minu-nmtf gauche main-**i** droit main-**wa** différent-St
(Minu, **la main gauche** est différente de la main droite)

Minu-nmtf deuxième tâche-**wa** premier tâche-**Acc** changer-Passé/St
(Minu a changé avec la deuxième la première tâche)

une ambiguïté structurale se crée à cause de l'identité de la postposition *-wa* du complément et la postposition de conjonction *-wa*. Autrement dit, (i) peut s'analyser d'une double manière :

(ii) a. N_0 $[N]_1$-**wa** $[N]_2$-*Acc V*
b. N_0 $[N$-**wa**$(=et)\ N]_1$-*Acc V*

Il en est de même pour :

(iii) 민우는 인아와 다르다
Minu-nîn Ina-wa talî-ta
Minu-nmtf Ina-**wa** différent-St
(Minu est différent de Ina)

qui permet l'ordre suivant des actants :

(iv) 인아와 민우는 다르다
Ina-wa Minu-nîn talî-ta
Ina-**wa** Minu-nmtf différent-St

La phrase (iv) peut s'interpréter soit comme une structure où le complément en *-wa* s'est déplacé en tête de phrase, soit comme une structure à sujet coordonné [*Ina*-**wa** *Minu*]. Pourtant cette double analyse n'est possible qu'avec la construction symétrique, car si l'on remplace l'adjectif de (iv) par un autre, non-symétrique, on n'observera pas la structure du type (iii). Soit :

(v) a. 인아와 민우는 뚱뚱하다
Ina-wa Minu-nîn t'ungt'ungha-ta
Ina-**wa** Minu-nmtf gros-St
(Ina et Minu sont gros)

b. *민우는 인아와 뚱뚱하다
**Minu-nîn Ina-wa t'ungt'ungha-ta*
Minu-nmtf Ina-**wa** gros-St
(*Minu est gros avec Ina)

La phrase (21a) présente une symétrie entre le sujet et le complément en *-wa*, et (21b) entre le complément en *-i* et le complément en *-wa*. Elles sont synonymes respectivement de :

=(22a) 인아는 민우와 성격이 다르다
Ina-nîn Minu-wa sôngkyôk-i talî-ta
Ina-nmtf Minu-**wa** caractère-**i** différent-St
(Ina est différente de Minu de caractère)

=(22b) 민우는 오른손이 왼손과 다르다
Minu-nîn olîn son-i ön son-kwa talî-ta
Minu-nmtf droit main-**i** gauche main-**wa** différent-St
(Minu, la main droite est différente de la main gauche)

Or, on observe une particularité de la phrase à complément N_2-*i* : la répétition d'un actant dans les deux autres est autorisée. Par exemple, pour (21a) on aura parallèlement :

(23) 민우의 성격은 인아의 성격과 다르다
Minu-îi **sôngkyôk**-*în Ina-îi* **sôngkyôk**-*kwa talî-ta*
[Minu-Gén caractère]-nmtf [Ina-Gén caractère]-**wa** différent-St
(**Le caractère** de Minu est différent de **celui** de Ina)

Et dans (21b), la redistribution d'un substantif n'est pas celle du complément en *-i*, mais celle du sujet (même si elle est fort redondante), puisque la symétrie existe entre les deux compléments :

(24) 민우의 왼손이 민우의 오른손과 다르다
Minu-îi *ön son-i* **Minu-îi** *olîn son-kwa talî-ta*
[Minu-Gén gauche main]-nmtf [Minu-Gén droit main]-**wa** différent-St
(La main gauche **de Minu** est différente de la main droite **de Minu**)

Ce type d'opération peut s'observer aussi dans certaines constructions verbales. Par exemple, avec le même classifieur *sôngkyôk* [caractère], on a la construction verbale restructurée :

(25a) 진오는 민우를 인아와 성격을 비교했다
Gino-nîn Minu-lîl Ina-wa **sôngkyôk-îl** *pikyoha-ôs'ta*
Gino-nmtf Minu-Acc Ina-**wa** **caractère-Acc** comparer-Passé/St
(Gino a comparé Minu avec Ina **de caractère**)

=(25b) 진오는 민우의 성격을 인아의 성격과 비교했다
Gino-nîn Minu-îi **sôngkyôk-îl** *Ina-îi* **sôngkyôk-kwa** *pikyoha-ôs'ta*
Gino-nmtf [Minu-Gén caractère]-**Acc** [Ina-Gén caractère]-**wa**
comparer-Passé/St
(Gino a comparé **le caractère** de Minu avec **celui** de Ina)

Cette observation nous fait supposer que la présence d'un troisième substantif dans la construction adjectivale n'est pas fondamentale et qu'il peut être une séquence détachée des deux autres actants par l'opération de *restructuration*.[5]

[5] Pour la définition de restructuration, voir A. Guillet et Ch. Leclère 1981.

On peut avancer un autre argument en faveur de cette hypothèse : c'est le caractère facultatif du complément en *-i* dans la phrase du type (21a). Autrement dit, on a aussi :

(26) 민우는 인아와 다르다
Minu-nîn Ina-wa talî-ta
Minu-nmtf Ina-**wa** différent-St
(Minu est différent de Ina)

En fait, le rapport entre ce troisième terme et les deux autres actants symétriques est un rapport **métonymique.**

Nous considérons donc que la construction de base des adjectifs symétriques ne comporte pas de troisième substantif :

N_0-*nmtf* N_1-**wa** *Adj-St*

et que le N_2-*i* sera obtenu par restructuration.

2.2. *N-i* est un substantif symétrique

La construction restructurée :

N_0-*nmtf* N_1-**wa** N_2-**i** *Adj-St*

est à distinguer d'un autre type de construction formellement identique :

(27) 민우는 진오와 관계가 깊다
Minu-nîn Gino-wa kwankye-ka kiph-ta
Minu-nmtf Gino-**wa** rapport-**i** profond-St
(Minu, son rapport est profond avec Gino)

Cette phrase diffère des précédentes, car :

- Le *N-i* y est obligatoire
Le rapport de symétrie n'est pas introduit par l'adjectif même, mais par le substantif du complément en *-i*. Ce substantif est donc nécessaire. On n'observe jamais :

*민우는 진오와 깊다
**Minu-nîn Gino-wa kiph-ta*
Minu-nmtf Gino-**wa** profond-St
(*Minu est profond avec Gino)

alors que l'effacement du complément en *-i* est possible dans la construction restructurée (e.g. (21a) = (26)).

- La répétition de *N-i* est interdite
Par ailleurs, la répétition de ce substantif dans les deux autres actants n'est pas non plus acceptable. Ainsi pour (27) on n'a pas :

*민우의 관계는 진오의 관계와 깊다
Minu-îi kwankye-nîn Gino-îi kwankye-wa kiph-ta
[Minu-Gén rapport]-nmtf [Gino-Gén rapport]-**wa** profond-St
(***Le rapport** de Minu est profond avec **celui** de Gino)

Pourtant, la phrase (27) présente bien un rapport de symétrie entre deux actants (*Minu* et *Gino*). Elle est synonyme de :

진오는 민우와 관계가 깊다
Gino-nîn Minu-wa kwankye-ka kiph-ta [6]
Gino-nmtf Minu-**wa** rapport-i profond-St
(Gino, son rapport est profond avec Minu)

C'est une construction où la symétrie est créée par un **substantif symétrique**[7] et non par un adjectif : ce substantif n'apparaît qu'avec la postposition *-i* dans la construction adjectivale. Et notons que ce substantif, associé à un verbe support, constitue la partie prédicative de la phrase : la symétrie ne s'établit donc qu'entre N_0 et N_1-*wa*.

---

[6] Cette phrase permet aussi une transformation du type :

(i) 진오는 민우와 깊은 관계이다
Gino-nîn Minu-wa kiph-în kwankye-i-ta
Gino-nmtf Minu-**wa** [profond-Sd rapport]-**ita**(être)-St
(Gino est en rapport profond avec Minu)

En fait, la séquence en *ita*, dans cette phrase, n'est pas de même nature que celle à copule *ita* qui introduit un attribut : *ita* dans (i) sera plutôt un terme support.

On observe d'autres phrases synonymes de (i), comprenant des termes supports comme *is'ta* (exister), *kacikois'ta* (avoir), qui demandent respectivement un complément en *-e* et un complément accusatif (en *-lîl*). Les phrases suivantes sont synonymes de (i) :

(ii) 진오는 민우와 깊은 관계에 있다
Gino-nîn Minu-wa kiph-în kwankye-e is'-ta
Gino-nmtf Minu-**wa** [profond-Sd rapport]-**e exister**-St

(iii) 진오는 민우와 깊은 관계를 가지고 있다
Gino-nîn Minu-wa kiph-în kwankye-lîl kacikois'-ta
Gino-nmtf Minu-**wa** [profond-Sd rapport]-**Acc avoir**-St

[7] En effet, le substantif symétrique ne se trouve pas seulement dans les constructions adjectivales, mais aussi dans les constructions verbales et les constructions à copule *-ita* (être) :

Le verbe qui accompagne le substantif symétrique est à distinguer de celui qu'on a observé dans (20). Comparons :

(i) 민우는 인아와 역할을 바꾸었다
Minu-nîn Ina-wa **yôkhal-îl** *pak'u-ôs'ta*
Minu-nmtf Ina-**wa rôle-Acc** changer-Passé/St
(Minu a changé **de rôle** avec Ina)

avec :

(ii) 민우는 인아와 계약을 맺었다
Minu-nîn Ina-wa **kyeyak-îl** *mäc-ôs'ta*
Minu-nmtf Ina-**wa contrat-Acc** établir-Passé/St
(Minu a établit **un contrat** avec Ina)

Les structures formelles étant identiques, on observe que si la symétrie dans (i) est créée par le verbe même *pak'uta* (changer), dans (ii) c'est par le substantif *kyeyak* (contrat), et le verbe y est un terme support(*Vsup*). Pourtant, les compléments en *-wa* et en *-i* sont obligatoires dans les deux cas. Les différences syntaxiques entre ces deux constructions nécessiteront une étude plus approfondie.

Une autre construction à substantif symétrique est celle à copule, par exemple :

(iii) 민우는 인아와 동창생이다
Minu-nîn Ina-wa tongchangsäng-i-ta
Minu-nmtf Ina-**wa** collègue-**ita**-St
(Minu est collègue avec Ina)

3. CONCLUSION

Nous avons observé deux types de constructions symétriques d'adjectifs :

- **la construction à adjectif symétrique**, définie par la relation :

(28) N_0-*nmtf* N_1-**wa** *W Adj-St*
= N_1-*nmtf* N_0-**wa** *W Adj-St*

- **la construction adjectivale à substantif symétrique**, définie par la relation :

(29) N_0-*nmtf* N_1-**wa** N_2-**i** *Adj-St*
= N_1-*nmtf* N_0-**wa** N_2-**i** *Adj-St*

où N_2 est un substantif symétrique obligatoire.

La relation (28) peut être définie comme une transformation qui s'applique à la construction à complément en *-wa*, qu'il s'agisse d'une construction verbale ou adjectivale : avec la phrase verbale cette transformation n'est pas une condition suffisante pour définir la construction symétrique, car le type de complément d'accompagnement en *-wa* subit cette opération sans introduire de rapport de symétrie avec le sujet.

Quant à la phrase adjectivale, la construction à complément en *-wa* acceptant cette transformation semble une condition suffisante pour déterminer la classe d'adjectifs symétriques, sauf dans le cas intermédiaire des adjectifs décrits en 1.2. (i.e. des adjectifs dont le complément en *-wa* alterne avec un complément datif en *-e*) où on n'a une phrase symétrique qu'en présence d'un pronom réciproque comme *sôlo* ("l'un Prép l'autre" ou "et réciproquement") : quand on insère *sôlo*, cette construction en *-wa* (dans ce cas-là, non-substituable par en *-e* datif) permet la transformation (28). Les caractéristiques particulières de ce cas intermédiaire, que nous appelons **construction pseudo-symétrique**, devront être étudiées plus en détail.

La même relation peut être projetée sur la construction adjectivale à substantif symétrique du type (29). Notons que s'il s'agit de la construction verbale, distinguer N_2-*Acc* symétrique de N_2-*Acc* qui joue un rôle d'actant sera un problème préalable. Cette opération transformationnelle, si elle est permise, y sera plus complexe et dépendra de conditions spécifiques. Ce sont des problèmes qu'on devra aborder dans les études sur la *symétrie* dans le lexique.

Pour l'instant, nous n'avons examiné que des adjectifs qui entrent dans la construction (28) : d'une part, les adjectifs qui permettent (29) sont, sans avoir le trait "symétrique", sémantiquement hétérogènes et constituent difficilement une classe. D'autre part, ils ont des contraintes sélectionnelles avec le substantif symétrique et non avec le sujet (e.g. pour un substantif *kwankye* (rapport), on aura des adjectifs comme *kiphta* (profond), *manhta* (riche), *côkta* (peu), *pokcaphata* (complexe) ou *thîkpyôlhata* (particulier), etc.). C'est une observation intéressante par rapport aux verbes qui accompagnent le substantif symétrique : ce sont plutôt des verbes supports comme *mäcta* (établir), *k'înhta* (rompre), *yucihata* (maintenir), etc.

Une dernière remarque : le choix lexical des actants joue un rôle important dans l'établissement du rapport de **symétrie** entre deux actants. Par exemple, dans la distribution suivante des actants, même avec un adjectif symétrique :

민우는 고릴라와 비슷하다
Minu-nîn kolilla-wa pisîsha-ta
Minu-nmtf gorille-**wa** ressemblant-St
(Minu est ressemblant à un gorille)

on n'observera pas de symétrie entre N_0 et N_1, alors qu'on l'a avec :

민우는 진오와 비슷하다
Minu-nîn Gino-wa pisîsha-ta
Minu-nmtf Gino-**wa** ressemblant-St
(Minu est ressemblant à Gino)

Pour qu'une phrase à adjectif symétrique soit une construction symétrique, les deux actants doivent être des arguments équilibrés. Mais, ce rapport est difficile à définir formellement. En fait, ce type de contrainte logique s'observe dans d'autres constructions. Par exemple, le verbe *phyônähata* [préférer] demande deux compléments sémantiquement équilibrés comme dans :

민우는 진오보다 인아를 편애한다
Minu-nîn Gino-pota Ina-lîl phyônäha-nta
Minu-nmtf Gino-**à** Ina-Acc préférer-St
(Minu préfère Ina à Gino)

même si ce verbe ne permet pas une construction symétrique :

=/= 민우는 인아보다 진오를 편애한다
Minu-nîn Ina-pota Gino-lîl phyônäha-nta
Minu-nmtf Ina-**à** Gino-Acc préférer-St
(Minu préfère Gino à Ina)

Une étude plus complète des propriétés syntaxico-sémantiques propres à la classe des *adjectifs symétriques* ainsi définie permettra de mettre en évidence sa spécificité. L'appartenance à des classes syntaxiques implique en effet, pour un mot, une série de propriétés qui caractérisent le fonctionnement du lexique dans la langue. Ces propriétés doivent donc figurer dans un dictionnaire syntaxique, électronique ou non, d'une manière systématique et exhaustive. Si l'étude qui précède est purement syntaxique, il ne faut pas perdre de vue, en effet, que les phénomènes sont lexicaux.

REFERENCES

BOONS, J.-P.; GUILLET, A.; LECLERE, Ch., 1976, *La structure des phrases simples en français / Constructions intransitives*, Genève ; Droz.

BORILLO, Andrée, 1971, Remarques sur les verbes symétriques français, *Langue française* 11, Paris ; Larousse.

GROSS, Maurice, 1975, *Méthodes en syntaxe*, Paris ; Hermann.

GROSS, Maurice, 1990, *Grammaire transformationnelle du français 3 - Syntaxe de l'adverbe*, Paris ; Asstril.

GUILLET, Alain ; LECLERE, Christian, 1981, Restructuration du groupe nominal, *Langages* 63, Paris ; Larousse.

HONG, Chai-Song, 1987, *Hyôntä hankukô tongsakumunîi yônku*(Etudes de constructions verbales en coréen cotemporain), Séoul ; Thap chulphansa.

NAM, Jee-Sun, 1990, Sur une construction N_0 N_1-*ita* en coréen, *Lingvisticae Investigationes*, XIV-2, Amsterdam ; J. Benjamins B.,V.

NAM, Jee-Sun, 1991, *Etablissement du corpus des adjectifs coréens*, Rapport technique N°30, Paris ; Institut Blaise Pascal.

Preparing a Text Corpus — Computational Tools and Methods for Standardizing, Tagging and Structuring Text Data

OLE NORLING-CHRISTENSEN

Most papers on corpora deal with criteria for the selection of text samples, or with methods for corpus analysis. This paper will go into the intervening process of transforming the selected text samples, which typically come from many different sources and follow many different standards, into uniform *corpus entries*, suitable for lexicographical analysis. It is described, how the text samples are being enriched with linguistic as well as extra-linguistic information, and it is underlined that clear-cut decisions on which *features* of the texts should be represented (e.g. by tagging), and which not, are necessary prerequisites for the processing. The paper pays special attention to the ongoing work on a 40 m. words corpus of modern Danish to be used by *The Danish Dictionary*. But the methods and tools described have a broader scope; some of them are, for instance, also used for converting printed dictionary texts into SGML-structured data.

The Danish Dictionary

On September 1st, 1991 works started on a new six volume dictionary of modern Danish. The project is carried out by The Danish Society of Language and Literature under the joint direction of *Ebba Hjorth*, *Iver Kjær* and the author. It is sponsored by the Danish Government and The Carlsberg Foundation (owner of most Danish and quite a few foreign breweries). The task of the editoral staff is to produce the best dictionary possible, under a fixed budget of 30 m. DKK (5 m. US$) and a fixed time schedule of eigth years: 18 months of preparatory work; 5 years of manuscript compilation; and 18 months of final work like proofreading and closing down.

The main sources for the dictionary are: *1)* a corpus of 40 m. words; *2)* a few important dictionaries; *3)* excerpts (roughly: 1 m.) and word lists collected since 1955 by the national advisory Board of the Danish Language; and, of course, *4)* the linguistic competence of the editors. The *corpus* does, in a balanced way, cover all kinds of text, including transcribed spoken language, from the period 1983-92. Among the *dictionaries*, special attention is paid to the official Danish spelling dictionary (RO, 1986), and two comprehensive bilingual dictionaries: Danish-English (V&B[3], 1990) and Danish-French (B&H[4], 1991). They all are available in different kinds of machine-readable form.

Even though the new dictionary is basically intended to be descriptive, spelling and inflexion must follow the official norm; this is what the spelling dictionary is used for. The first editions of the two bilingual dictionaries were both based on the 28 volume Dictionary of the Danish Language (ODS, 1918-56) that covers the language from 1700 to c. 1915/55. They basically use the same meaning discriminations as the ODS; but during three (V&B[3]) or four (B&H[4]) revisions they have, of course, modernized the vocabulary and the stock of collocations and idioms. They are, thus, the most

outstanding representatives of a living Danish lexicographical tradition to be continued by The Danish Dictionary.

Computers are being intensively used in all phases of the project. During the present preparatory period (September 1991 - March 1993) corpus texts are being scanned or keyed in or converted from all kinds of wordprocessing or typesetting formats; and information on author(s), text type etc. is attached, more or less automatically, to each selected piece of text. The possibility of automatic (syntactic and) morphological tagging of the corpus is explored, but not yet decided on. Only a rudimentary word class tagging has been made, as proper names and abbreviations are tagged as such during the process of dividing the written texts into sentences. In parallel with this, word lists are built and gradually enriched into semi-manufactured "skeleton entries" that comprise, in a structured manner, as much information as it has been possible to extract from the dictionaries mentioned. Before and during the editorial period (April 1993 - March 1998) the relevant corpus evidence for each word will be searched, analyzed, structured and attached to the skeletons. The editors' job, then, is to change the skeletons into finished entries. For the editorial work, the SGML-based system GestorLEX is used. It has been developed by the Danish software house TEXTware A/S in close collaboration with the author, and it meets most of the requirements put forward in (DANLEX 1987: 239-251). Further, a powerful dedicated corpus program, to be used in connection with GestorLEX, is currently in process of construction.

Machine-Readable / Machine-Usable

Today, many texts can be obtained as data files from typesetting systems or from the authors' wordprocessing equipment. Others must be transformed into data files by scanning a printed version or by transcribing (by word processor) a tape recorded version. In any of these cases, however, the data must be transformed into one uniform and consistent format suitable for those computational searchings and analyses that will be the future use of the corpus. Defining such a format is no trivial task. It implies a series of decisions on *which features* of the text you want to depict in the machine-readable version. The irrelevant features should be left out, while the relevant ones must be rendered, *e.g.* by tagging, in a uniform and unambiguous way. Besides, a character set standard (Code Page) must of course be laid down.

Should there, for instance, be specific codes for the smell of the newspaper or the quality of its paper? - probably not. The colour of the paper? - it might have some specific meaning. The size of the letters? - differences in size are likely to signify differences in text type; but the meaning of such differences will differ from one text to another. An obvious conclusion from this kind of questions is that the coding has to be generic and not just mirroring how the printer chose to represent the different kinds of text: *Business pages*, not *Pink paper*; *Headline*, not *Big bold type*. The Text Encoding Initiative (TEI 1990; TEI 1991) has defined an abundance of generic tags to account for most kinds of features of most kinds of text. In praxis, however, corpus builders will have to select only a very restricted number of features of the original books, newspapers, etc. for representation in the corpus.

Representing spoken language

The objects clause for The Danish Dictionary states that the dictionary shall *cover* the written language and *consider* the spoken language. The reason for this unbalance is obvious: the total of Danish (and every other) spoken language is much more difficult to define and to represent in a corpus. We do not have enough ressources (neither time nor money) to collect and transcribe very much spoken language ourselves. Instead, we have tried to find all kinds of transcriptions already existing, and a rather big material has come up. In many cases we also have got access to the original tape recordings or sound tracks, which enables us to check the quality of the transcriptions and correct them if necessary. The

transcriptions have been made by different persons and for different purposes: linguistic, psychological or sociological research, or just as a documentation of what was said in radio and television, or in the Danish Parliament and the City Council of Copenhagen. This implies that a lot of different conventions are used for notating especially features like pauses, cleft sentences, laughter, unintelligible passages, the transcriber's explanatory comments. To deal with all this, it is, again, important to make clear-cut decisions on which features should and should not be represented in the machine-readable version of the speech.

A standard format for corpus samples

The international standard SGML (ISO 8879, 1986) for generic description of textual structures and for marking up the texts accordingly, is used by The Danish Dictionary for describing and tagging not only the dictionary but also the corpus. Readers who are not familiar with SGML and with terms like DTD, element, attribute, entity reference, may consult the brilliant introduction in (TEI 1990: 9-32).

For the corpus an SGML document type *CorpusEntry* has been defined. It provides a suitable form for registration of the necessary (extralinguistic) information about the text as well as a means for unambiguous tagging of those (linguistic) features of the text proper that we have decided to represent in the corpus. Each sample (element *CorpusEntry*) consists of: A *Header*, that contains information on the kind and provenience of the text, and the *Text* proper. In the language of SGML:

```
<!DOCTYPE CorpusEntry [
<!ELEMENT CorpusEntry    ( Header, Text ) > ] >
```

The full DTD (Document Type Definition) of a corpus entry is given in the Appendix.

The Header

For designing the Header and deciding which information types should form part of it, we found much inspiration in (Atkins 1991). The Header of each corpus entry is divided into two main parts, *viz.* information on the source (SourceInfo), and information on the text sample proper (TextDescription). *SourceInfo* consists of an unambiguous identification (TextGroup + TextNumber), notes on restrictions of use imposed by the supplier of the text (private or confidential texts), information on those who produced the text (LanguageUser = authors, speakers), and on title, publisher, date of origination, and location (*e.g.* page number). There is one element *LanguageUser* for each person involved in the production of a given text sample. Especially in spoken language there is usually more than one. The element describes the person's name, role (*e.g.* interviewer or interviewee), sex, education, occupation, year and place of birth, and language variant (*i.e.* standard or regional Danish). The element *TextDescription* gives an account of the language type (general or special), whether it is written or spoken, and public (reception) or private (production), the age relation between sender and receiver of the text (adult-adult, adult-juvenile, adult-child, juvenile-adult, etc.), medium (book, newspaper, television etc.), genre, subject field, size of the sample.

The full structure and contents of the header can be explained in the following way (cf. the DTD of the Appendix). An interrogation mark (?) after an element name means that the element is facultative, *i.e.* it shall only be there if it is relevant, and if the information in question is known. The plus (+) after LanguageUser means that ther may be one or more of these element in a single header.

Header
  SourceInfo
    TextGroup *Unambiguous identifier of a group of (related) corpus entries*
    TextNumber *Serial number inside the text group*
    Restriction?
      RestrictA *Proper names in text must be altered: "Y[es]"/"N[o]"*
      RestrictB *Text must only be used for the dictionary: "Y[es]"/"N[o]"*
      Expiration *of Restriction B, e.g. "1998"*
    LanguageUser+ *(one element for each author/speaker)*
      Role? *Esp. when more language users are involved; e.g. "teacher", "pupil"*
      Identification? *A unique three character string, referred to by SpeakerTurns in the Text*
      LastName? *If known*
      FirstName? *If known*
      ***Sex** "m"/"f"/"u[nknown]"*
      Education? *if known*
      Occupation? *if known*
      ***YearOfBirth** a number between 1880 and 1990*
        Precise? *"?", if not known exactly*
      PlaceOfBirth? *if known*
      ***LanguageVariant** "standard"/"regional"*
    TextTitle? *if any*
    VolTitle? *Name of Anthology, Newspaper, Magazine, etc., if any*
    Publisher? *Book publisher or Radio or TV station, if any*
    Date
      Day? *if known*
      Month? *if known*
      Year *number between 1983 and 1992*
        Precise? *"?", if not known exactly*
    Location? *e.g. Section, page, column of Newspaper; (Vol.,) page of book*
  TextDescription
    ***LanguageType** "general"/"special purpose"*
    ***Written_Spoken** "written"/"spoken" or one of two intermediate types*
    ***Aspect** "reception"/"production"*
    ***AgeRelation** "child-child"/"child-juvenile"/"child-adult"/../"adult-adult"/"unknown"*
    ***Medium** taken from a list of 12 different media, e.g. book, journal, radio, film*
    ***Genre?** taken from a list of 124 partly medium-dependent genres, like novel, letter, comic*
    ***Subject?** taken from a list of 64 different subject areas, like biology, literature, physics*
    Size *Number of words (tokens) in this text sample*

The elements marked by an asterisk (*) above are standardized descriptors that play a special role in corpus search and analysis. For each of the descriptors a restricted list of legal values is defined. Different text types, and corresponding subcorpora, can be defined in terms of one or more of these descriptors, *e.g.* "Women born before 1940 speaking to children" or "Newspaper texts on politics". Besides, the decriptors are used for studies of the distribution of all kinds of linguistic features over the different text types.

The Text

The structure of the *Text* element depends on whether it consists of written language or of (transcribed) spoken language. Written language is split up into paragraphs (the element *p*) that are subdivided into sentences (the element *s*). Sentences are mostly non-tagged strings of characters (the

SGML category #PCDATA); these may, however, be interspersed with elements of special types of text, *viz.* the elements *Highlighted, Note, PropName* and *Abbrev*. The tag *Highlighted* covers all kinds of accentuation in the original text: underlining, boldface, italics, spacing, bigger or deviant founts; *Note* are foot- or endnotes.

Spoken language is normally not cut into paragraphs; instead, they may be divided up into speaker turns. Most of the spoken texts are conversations or interviews with more persons involved. Consequently, the header contains two or more instances of the element *LanguageUser*. Each of them contains in the subelement *Identification* a different three letter string. Each element *SpeakerTurn* contains an attribute *id* that refers to the *Identification*. The *SpeakerTurn* element consists of #PCDATA interspersed with entity references like {hesitation} representing non-verbal sounds like 'eh', 'mmm'; {pause}; {uf} that represents a passage that was incomprehensible to the transcriber; {laughter}; and with the elements *Comment* (the transcriber's "stage directions" that are not part of the speech), and *Doubtful*: a word or passage that the transcriber was not sure about.

Computational tools

As much as possible of the Header-information, as well as the identification and tagging of the entity references and subelements of the Text proper, is made (semi)automatically. This means, that for each group of texts of a given provenience or type, a tailor made conversion program is written. Not only does the program convert a given wordprocessor format into our standard format; in some cases it also makes use of the authors' idiosyncratic ways of marking those features we are interested in. In other cases these features are marked up manually, using word processor macros.

The most important software used for the preparation of the machine-readable sources for the dictionary are: the context free chart parser DIPA written in C by Peter Molbæk Hansen, assistant professor in computational linguistics at the University of Copenhagen; the general text conversion system DICONV written in Turbo Pascal (object oriented versions 5.5 / 6.0) by the author; and the Paradox Engine of Borland International Inc. Besides, a standard word-processor (WordPerfect) and database management system (Paradox) are used.

These tools, in different combinations, are utilized not only for the preparation of corpus entries, but also for making all kinds of wordlists, and for their gradual extension into skeleton entries.

DIPA (*Di*ctionary *Pa*rser) was originally made for the author's structure analysis and SGML tagging of dictionary data (Norling-Christensen 1992). At the Danish Dictionary it is especially used for processing those dictionaries that are part of our sources, and to analyze and tag Header information. As input DIPA takes two or three files: The text that is to be analyzed, e.g. the typesetting data for a printed dictionary; A grammar (list of rewriting rules) that defines the syntax to be applied; And, optionally, a lexicon, *i.e.* a list of strings with an attached class-marker (eg the dictionary's abbreviation list, the abbreviations being classified as part-of-speech markers, register labels, subject classifications, etc.). Four files are output: A file of accepted entries; a file of rejected entries (that did not match the grammar); an SGML-tagged file (only the accepted entries); and a message file with information on how many entries were processed, accepted and rejected, the time of start and end of the parsing, and some technical information that can be used for optimizing the program. DIPA can be run in batch mode or interactively. In the interactive mode, when an entry is rejected, one can choose to edit the grammar, the lexicon, or the input entry itself; or to continue with the next entry. This is very useful when a new grammar is being developed, as well as for correcting syntactical errors in the input text.

DICONV (*Di*ctionary *Con*verter) was originally made for converting typesetter files into word

processor files and vice versa, and for making all kinds of automatic correction and change. Later on it has been used as a pre- and postprocessor to DIPA, and facilities for treating the tree structures of SGML-tagged data have been added. At the Danish Dictionary it is particularly being used for processing text samples for the corpus. DICONV itself is not a program, but a library of Turbo Pascal Units. A DICONV program consists of a series (up to 20) modules ("transducers") that are run after each other, the output of one module being the input for the next one. In the heart of each module is a rather short piece of Pascal code (a virtual procedure), that draws on library procedures for looking backwards and ahead, looking up in one or more tables, submitting error messages (the error conditions being defined by the programmer), treating SGML tags, attributes, and text separately, etc. Besides the converted output file, a report file is produced. It records the names of the input files, the time used and the number of characters processed; further, all errors encountered are reported with a few lines of context around the erroneous spot. This program, too, can be run interactively: Independently of the others, each module can run in one out of three modes that are initially set by the programmer: EditAll, EditOnError or NoEdit; these settings can be changed at any time during a run. When a module turns interactive after having processed every section of input text (EditAll) or after having encountered errors (EditOnError), the screen turns into two text editors showing the input and the output of the module in question, and the error message if any. The input or the output can then be corrected; if the user chooses to correct the input, the corrected input will be re-processed. Extensive reuse of modules makes the programming fast and reliable.

Paradox Engine is a applications programming interface for Pascal and C. It gives the programmer access to all kinds of basic database functions like creating tables with one or more indexes, searching, reading, writing etc. The programs can be used in networks and other shared environments, as the necessary locking and unlocking of records etc. is supported. The file formats are the same as those used by Borland's database management system *Paradox*, which means that the same data kan be used and manipulated both by Paradox and by ones own programs. For the computer assisted creation of the headers of the corpus entries we use a custom-built Paradox application displaying simultaneously one record from each of three temporary tables: A LanguageUser table; a table with the rest of the header information; and the text of the corpus entry. This screen form is used for keying in those parts of the header information that have not been entered automatically. The three tables are made by our programmer with the help of Turbo Pascal and the Paradox Engine. For each group of text, as much header information as possible is entered into the table before it is handed over to the editor who shall complete the header. Most often, the automatically entered data is the information that is common to a batch of text. They may *e.g.* come from the same source, cover the same subject field, be of the same medium, genre or language type. In some cases, however, much more information can be automatically entered. This is especially the case of such texts that already have been classified and described by others. Such texts are available from other, minor, corpus projects and from a newspaper information system, the data of which we have received on magnetic tape. When the header has been completed, the temporary tables are emptied and their contents is stored in SGML-format. The header information, but not the texts, is also transfered to a permanent database that gives us a good grasp of the corpus and its different text types.

Perspectives

The corpus described has as its object a dictionary of contemporary Danish general language. As a consequence, LSP - the language produced for specialists by specialists - is not included. The corpus covers the period 1983-92, and texts newer than that will not be included, at least not by this dictionary staff. Deliberately we have restricted ourselves to include in the headers, and in the tagging of the texts, only such information which we regard useful for the dictionary job. Researchers in other fields might have wished for more types of information and other kinds of tagging. In spite of these limitations we expect, however, that our corpus will be in great demand by other researchers of

Danish, as it is by far the biggest Danish text corpus, and the only one to include nearly all text types. Further, those kinds of information that are included, are rendered in a consequent and well documented way. Except for rival projects (if any), the text corpus of the Danish Dictionary will, therefore, be made accessible to other researchers, and it is our hope that a way will be found to reuse the methods, tools and principles developed, for further corpus building.

Literature

Atkins 1991: Atkins, Sue, Jeremy Clear & Nicholas Ostler: Corpus Design Criteria. 8th November 1991. To appear in *Literary and Linguistic Computing*.

B&H[4] 1991: Blinkenberg, Andreas & Poul Høybye: Dictionnaire Danois-Français/Dansk-fransk Ordbog. 4th edition. Ed. Jens Rasmussen & al. Vol. 1-2. Copenhagen 1991.

DANLEX 1987: The DANLEX Group (Ebba Hjorth, Jane R. Jacobsen, Bodil Nistrup Madsen, Ole Norling-Christensen, Hanne Ruus): Descriptive Tools for Electronic Processing of Dictionary Data. Studies in Computational Lexicography. Lexicographica Series Maior 20. Tübingen 1987.

ISO 8879 1986: International Organization for Standardization: Information processing - Standard General Markup Language (SGML). [Geneva]: ISO, 1986.

Norling-Christensen 1992: Struktureret redigering af Ordbøger [Structured Editing of Dictionaries]. In Ruth Vatvedt Fjeld (ed.): *Nordiske Studier i Leksikografi. Rapport fra Konferanse on Leksikografi i Norden 28.-31. mai 1991.* Oslo 1992: 447-454.

ODS 1918-56: Ordbog over det Danske Sprog [Dictionary of the Danish Language]. Udgivet af Det Danske Sprog- og Litteraturselskab [ed. by The Danish Society of Language and Literature]. Vol 1-28. Copenhagen 1918-56.

RO 1986: Dansk Sprognævn [Board of the Danish Language]: Retskrivningsordbogen [The Spelling-Dictionary]. Copenhagen 1986.

TEI 1990: Burnard, Lou, & C.M. Sperberg-McQueen (ed.s): Guidelines For the Encoding and Interchange of Machine-Readable Texts. Document Number: TEI P1. Text Encoding Initiative, Chicago, Oxford. ACH, ACL, ALLC. Draft: Version 1.1, October 1990.

TEI 1991: Burnard, Lou, & C.M. Sperberg-McQueen: Living with the Guidelines. TEI EDW21: An Introduction to TEI Tagging. The first TEI Europaean workshop 1-2 July 1991. Oxford University Computing Service 24 Jun 1991.

V&B[3] 1990: Vinterberg, Hermann & C.A. Bodelsen: Dansk-engelsk Ordbog. 3rd edition. Ed. Viggo Hjørnager Pedersen. Copenhagen 1990.

Acknowledgements

The corpus design described above has been made jointly by Ebba Hjorth, joint managing editor, Kjeld Kristensen, senior editor, and the author. I thank my two colleagues for our many fruitful discussions, and for their comments to this paper.

Appendix
Document Type Definition, The DDO Text Corpus Entry

```
<!DOCTYPE CorpusEntry [
<!ELEMENT CorpusEntry ( Header, Text ) >

-- The Header --

<!ELEMENT Header ( SourceInfo, TextDescription ) >

<!ELEMENT SourceInfo  ( TextGroup, TextNumber, Restriction?, LanguageUser+, TextTitle?,
                        VolTitle?, Publisher?, Date, Location? ) >

<!ELEMENT ( TextGroup, TextNumber ) ( #PCDATA ) >

<!ELEMENT Restriction ( RestrictA, RestrictB, Expiration) >

<!ELEMENT ( RestrictA, RestrictB, Expiration) ( #PCDATA ) >

<!ELEMENT LanguageUser  ( Role?, Identification?, LastName?, FirstName?, Sex?, Education?,
                          Occupation?, YearOfBirth?, PlaceOfBirth?, LanguageVariant ) >

<!ELEMENT     ( Role, Identification, LastName, FirstName, Sex, Education, Occupation,
                PlaceOfBirth, LanguageVariant ) ( #PCDATA )>

<!ELEMENT YearOfBirth ( #PCDATA, Precise? ) >

<!ELEMENT ( TextTitle, VolTitle, Publisher, Location ) ( #PCDATA )>

<!ELEMENT Date ( Day?, Month?, Year, Precise? ) >
<!ELEMENT ( Day, Month, Year, Precise ) ( #PCDATA )>

<!ELEMENT TextDescription   ( LanguageType, Written_Spoken, Aspect, AgeRelation, Medium,
                              Genre?, Subject?, Size )>

<!ELEMENT     ( LanguageType, Written_Spoken, Aspect, AgeRelation, Medium, Genre,
                Subject, Size ) ( #PCDATA ) >

-- The Text --

<!ELEMENT Text                ( p+           | (SpeakerTurn | Comment )+ ) >
                    -- written paragraphs | speech transcriptions --

<!ELEMENT p (s+) >   -- paragraph = sentences --

<!ELEMENT s ( #PCDATA ) +(Highlighted, Note, PropName, Abbrev) >

<!ELEMENT SpeakerTurn ( #PCDATA ) +( Doubtful, PropName) >

<!ELEMENT Comment ( #PCDATA )>

<!ELEMENT Highlighted ( #PCDATA ) -(Highlighted) >

<!ELEMENT Note ( #PCDATA ) -(Note) >

<!ELEMENT PropName ( #PCDATA ) -(PropName) >

<!ELEMENT Abbrev ( #PCDATA ) -(Abbrev) >

<!ELEMENT Doubtful ( #PCDATA ) -(Doubtful) >

<!ATTLIST SpeakerTurn id NUTOKEN #IMPLIED > ] >
```

Compiling Dictionaries with Grammar Defined Databases

JÚLIA PAJZS — LÁSZLÓ TIHANYI — ILDIKÓ VILLÓ

The notion of grammar defined databases becomes more and more well known, because it is the most convenient method for compiling and storing electronic dictionaries. This paper descibes two applications: the new French/Hungarian - Hungarian/French Dictionary and the Historical Dictionary of Hungarian, both of them using the same program for checking and typesetting the dictionary entries during compilation of the dictionaries.

Recently the use of grammar defined databases for storing dictionary entries became widespread. In a grammar defined database the structure of the database is defined by a grammar where one can only store entries which satisfy this grammar. Because the dictionary entries are varies to a great extent both in length and structure, this way of handling the electronic dictionaries is much more convenient for text storing than the traditional relational databases. It is possible to define all of the possible variants of the entries by a grammar. This is flexible way of storing dictionary entries which keeps the necessary information for the retrieval and conversion of the data. As Gonnet and Tompa (GONNET-TOMPA 1987) have shown, the grammar defined databases are as easy to handle as the relational ones, and they give a much more natural way for storing text.

The two projects which we will present here are not only similar in the way they will store the dictionaries to be compiled, but they also use the same corpus as source material. The project for compiling the Historical Dictionary of Hungarian has started six years ago in the Research Institute for Lingusitics, Hungarian Academy of Sciences (PAJZS 1990, PAJZS 1991). The idea was to compile a dictionary of Hungarian based on historical principles by use of a computerized corpus. At the outset we planned to collect about 13 million running words for covering the vocabulary of Hungarian from the earliest Hungarian prints up to the most current printed material, but after recording the 19th century corpus, which contains about 7 million words we had to shift the concept of our project. We realized that we needed a much larger corpus for the last five centuries, and we also had to cope with the fact that the computerized corpus alone can not be a basis of a "real" historical dictionary. Therefore we have decided to compile the dictionary of the last two centuries first (1772-1990), by using a much larger corpus (containing at least 20 million running words for this period). We also have about 5 million old fashioned dictionary slips for these two centuries, which were collected between 1884-1960 for the same purpose. Our lexicographer who compiles the draft entries finds combining this two kind of source material particulary useful. So far more than half of the planned corpus is recorded.

After keyboarding, the running texts are analyzed morphologicaly by a program (see in PRÓSZÉKY - TIHANYI 1992) which segments the running words to stems and inflections. The analyzed text will be retrieved by a text searching program (possibly by PAT: GONNET 1987), and the lexical entries are compiled using the lemmatized concordances.

The project for compiling a new French/Hungarian - Hungarian/French Dictionary has started last year at the Universite Paris III, Centre Interuniversitarire d'Etudes Hongroises in collaboration with the Centre Interuniversitaire d'Etudes Francaises, Budapest and the Jozsef Attila University, Szeged. Since the last French/Hungarian - Hungarian/French dictionaries were made about 40 years ago, this will be a brand new one based on newly published Hungarian bilingual dictionaries, recent French monolingual dictionaries and also on a computerized corpus. This contains those prose excerpts which were collected for the Historical Dictionary of Hungarian and were published after 1960. This project also use a large corpus of recently published texts from newspapers. Their corpus contains about 1.5 million running words.

Both dictionaries will also use a transcribed spoken text corpus which was originally collected for the Survey of Spoken Hungarian.

In both cases the dictionary entries are written using a normal text editor, rather than using any word-processor specific facilities, the different parts of the entry will be tagged by a modified version of SGML (Standard Generalized Markup Language). After keyboarding the entries a LEX program checks if the article satisfies the grammar, and if it does, the program writes typesetting commands instead of the SGML symbols. Therefore the dictionary will be kept in two disjunct formats: the database in SGML format and the text to be printed in typeset format. The typeset format is easier to proofread and to correct, the SGML format is more usable for the computerized version. In the SGML format the special Hungarian and French accented characters are coded by a combination of letters and numbers.

The grammar of the French/Hungarian - Hungarian/French Dictionary is the folllowing:

```
DIC       :=  [ART]+
ART       :=  ENT  [BGR]* [BLS]+ [LFG] [IFS] COL
ENT       :=  VDT [PHO] [MOR] CGR [RCT] [MAE]
MAE       :=  ([DDS] | [NDL] | [LIG])
BGR       :=  rszam CGR [BLS]+ [LFG] [IFS]
BLS       :=  [szam] [RCT] [MAE] [IDS] [EQV]+ [EXP]*
EQV       :=  [IDS] mots [CGR] [MAE]
EXP       :=  mots [MAE] TRD
TRD       :=  mots [MAE]
LFG       :=  [EXP]+
mots      :=  [a..z A..Z 0..9 .,;!?'"]
szam      :=  [0-9]
rszam     :=  [IVXL]
COL       :=  NOM DAT
NOM       :=  NOK NOP
DAT       :=  DAP DAF
```

Where the beginning and end of fields are marked with the symbols:

DIC:= <DIC>.....</DIC>	dictionnaire
ART:= <ART>.....</ART>	article
ENT:= <ENT>.....</ENT>	entrée
BGR:= <BGR>.....</BGR>	bloc grammaticale
BLS:= <BLS>.....</BLS>	bloc sémantique
PHO:= <PHO>.....</PHO>	phonetique
MOR:= <MOR>.....</MOR>	morphologie
LFG:= <LFG>.....</LFG>	locution figée
IFS:= <IFS>.....</IFS>	informations supplémentaires
VDT:= <VDT>.....</VDT>	vedette
CGR:= <CGR>.....</CGR>	catégorie grammaticale
DDS:= <DDS>.....</DDS>	domaine de spécialité
NDL:= <NDL>.....</NDL>	niveau de langue
LIG:= <LIG>.....</LIG>	limitation géographique
RCT:= <RCT>.....</RCT>	constr. gramm. spécifiques
IDS:= <IDS>.....</IDS>	indications de sens
EQV:= <EQV>.....</EQV>	équivalents
EXP:= <EXP>.....</EXP>	exemple
TRD:= <TRD>.....</TRD>	traduction
NOK:= <NOK>.....</NOK>	nom de collaboreteur
NOP:= <NOP>.....</NOP>	nom d'operateur
DAP:= <DAP>.....</DAP>	dat premiere
DAF:= <DAF>.....</DAF>	dat finale

The dictionary entries are written using the Wordperfect editor. The accented characters are written in Wordperfect format and are converted by a program to a normal ASCII file where the accented characters are represented by a combination of letters and numbers. An example of the tagged articles in the modified SGML format:

```
<ART>
 <ENT>
  <VDT>1 fal</VDT>
    <CGR>v tr/intr</CGR>
 </ENT>
 <BLS>
    <EQV>delvorer; manger goulu5ment:</EQV>
    <EXP>mielrt falsz, egyell rendesen!
        <TRD>arre5te de t'empiffrer, mange correctement;</TRD>
    </EXP>
    <EXP>a gelpkocsi falja a kilomeltereket
        <TRD>la voiture delvore les kilome4tres;</TRD>
    </EXP>
    <EXP>szerelmesregelnyeket fal
        <TRD>delvorer des romans d'amour</TRD>
    </EXP>
  </BLS>
<NOK>szeptemberi csapat</NOK>
<NOP>chantal</NOP>
<DAP>18 janvier 1992</DAP>
<DAF>14 jullius 1992</DAF>
</ART>

<ART>
  <ENT>
      <VDT>2 fal</VDT>
          <CGR>n</CGR>
  </ENT>
  <BLS>
  <EQV>mur
        <CGR>m;</CGR>
  </EQV>
  <EQV>
```

```
      <IDS>(veldo3fal)</IDS>
            muraille
                <CGR>f;</CGR>
  </EQV>
  <EQV>
   <IDS>(ko2zfal)</IDS>
          cloison
             <CGR>f;</CGR>
  </EQV>
  <EQV>paroi
             <CGR>f;</CGR>
  </EQV>
  <EQV>
      <IDS>(edelny;elr)</IDS>
           paroi
              <CGR>f:</CGR>
  </EQV>
  <EXP>falat emel/hulz
      <TRD>dresser/ellever un mur;</TRD>
  </EXP>
  <EXP>a falnak talmaszkodik
       <TRD>s'appuyer contre le mur</TRD>
  </EXP>  </BLS>
  <LFG>
     <EXP>nelgy ~ ko2zo2tt
          <TRD> entre quatre murs;</TRD>
     </EXP>
     <EXP>ostromlolk fala
          <TRD>le mur/la vague des assaillants;</TRD>
     </EXP>
     <EXP> a falat talmasztja
          <TRD> se tourner les pouces;</TRD>
     </EXP>
     <EXP>falba u2tko2zik
          <TRD> se heurter a4 un mur;</TRD>
     </EXP>
     <EXP> a falba veri a fejelt
          <TRD> se taper la te5te contre les murs;</TRD>
     </EXP>
     <EXP> ezt falboll teszi
         <NDL>arg</NDL>
              <TRD> c'est de la frime;</TRD>
     </EXP>
     <EXP>falhoz alllilt
          <TRD>mettre au pied du mur;</TRD>
     </EXP>
     <EXP> a falnak beszell
          <TRD> parler aux murs</TRD>
     </EXP>
     <EXP> fejjel megy a falnak
          <TRD> foncer dans le tas
                        <NDL> fam;</NDL>
          </TRD>
      </EXP>
      <EXP> a falnak is fu2le van
           <TRD> les murs ont des oreilles;</TRD>
      </EXP>
      <EXP> ez falra halnyt borsol
           <TRD> autant parler a4 un mur; c'est un coup d'elpele  dans
l'eau;</TRD>
      </EXP>
      <EXP> etto3l falra malszik az ember
           <TRD> c'est a4 se taper la te5te contre le(s) mur(s) </TRD>
      </EXP>
  </LFG>
<NOK>szeptemberi csapat</NOK>
<NOP>chantal</NOP>
<DAP>18 janvier 1992</DAP>
<DAF>17 julius </DAF>
</ART>
```

This article is parsed by a program generated with YACC/LEX. If the entry does not satisfy the above grammar, the program gives a syntax error message, otherwise it creates an output file where the tages are replaced by Wordperfect typeface codes, and the accented characters are replaced by the Wordperfect extended characters. After the conversion the article is in printable Wordperfect format:

1 fal *v tr/intr* dévorer; manger goulûment: **miért falsz, egyél rendesen!** arrête de t'empiffrer, mange correctement; **a gépkocsi falja a kilométereket** la voiture dévore les kilomètres; **szerelmesregényeket fal** dévorer des romans d'amour
(szeptemberi csapat,chantal: 18 janvier 1992; 14 július 1992)
2 fal *n* mur *m; (védőfal)* muraille *f; (közfal)* cloison *f;* paroi *f; (edény;ér)* paroi *f:* **falat emel/húz** dresser/élever un mur; **a falnak támaszkodik** s'appuyer contre le mur **négy ~ között** entre quatre murs; **ostromlók fala** le mur/la vague des assaillants; **a falat támasztja** se tourner les pouces; **falba ütközik** se heurter à un mur; **a falba veri a fejét** se taper la tête contre les murs; **ezt falból teszi** *arg* c'est de la frime; **falhoz állít** mettre au pied du mur; **a falnak beszél** parler aux murs **fejjel megy a falnak** foncer dans le tas *fam;* **a falnak is füle van** les murs ont des oreilles; **ez falra hányt borsó** autant parler à un mur; c'est un coup d'épée dans l'eau; **ettől falra mászik az ember** c'est à se taper la tête contre le(s) mur(s) (szeptemberi csapat,chantal: 18 janvier 1992; 17 julius)

Presently, the grammar of the Historical Dictionary of Hungarian is somewhat simpler, but so far have only written draft entries for this dictionary.

```
SZOTAR      := [SZO]+
SZO         := [FEJ] [JEL]+
FEJ         := CIM [MIN] [VLT]
JEL         := [RSZAM] [GR] SZOF [FO]+ [AL]* [ETIM]
FO          := [SZAM] [SZOF] [GR] [MIN] [INF] [VON] [FR] [ERT]
[PLD]+ [VO]
AL          := BETU FO
ERT         := [SPEC] [EXP] [SPEC]
PLD         := MON FOR
```

Here the start and the end of the fields are marked the same way as in the other dictionary, and the meanings of the abbreviations are the following:

```
SZOTAR      := dictionary
SZO         := entry (article)
FEJ         := head of the entry
JEL         := semantic unit
RSZAM       := roman number
GR          := grammatical restrictions
SZOF        := part of speech
FO          := meaning
AL          := specialized meaning
ETIM        := etymology
SZAM        := number of sense
INF         := lexical- grammatical informations
VON         := obligatory government
FR          := phraseological unit
ERT         := definition
SPEC        := restriction of sense
```

A very short draft entry of the historical dictionary with these symbols looks like the following:

```
<SZO>
     <FEJ>
          <CIM> konzulta11 </CIM>
     </FEJ>
     <JEL>
          <GR>tn</GR>
          <SZOF>ige</SZOF>
          1.
          <MIN>Orvos Isk</MIN>
          <ERT>Konzulta1cio1t tart.</ERT>
     </JEL>
     <JEL>
          2.
          <ERT>Vmilyen ke1rde1st szake1rto3vel megbesze11. </ERT>
     </JEL>
</SZO>
```

These entries will be converted to more readable files the same way as the bilingual entries. The database will be kept again in the SGML format, the converted version will be used for reading and correcting the already completed dictionary entries.

In both projects only the first draft entries are written now, so we do not have much experience in the use of this system. Therefore it should only be regarded as an experimental version. It seems a relatively easy way for writing SGML databases, but it might not be comfortable enough for the keyboarders of the dictionary entries. Right now we are planning to develop it further so as to make it more comfortable and safe to use. On the other hand we are trying to get experience with other software on the market which are developed for writing SGML databases (GestorLex, WriterStation for example). One thing is certain, these dictionaries will be written directly in SGML format, and they will be probably sold on floppy disk in database format as well.

Bibliography:

GONNET, G.: (1987) *PAT - An efficient text searching system* University of Waterloo Centre for the New OED.

GONNET, G. - TOMPA, F.: (1987) *Mind your Grammar: a New Approach to Modelling Text.* University of Waterloo Centre for the New OED.

PAJZS J.: (1990) Creating a Historical Dictionary of Hungarian with the Aid of Computer *T. MAGAY - J. ZIGANY: BUDALEX '88 Proceedings* Akadémiai Kiadó Budapest p. 559-563.

PAJZS J.: (1991) The Use of a lemmatized Corpus for Compiling the Dictionary of Hungarian In: *Using Corpora Proceedings of the 7th Annual Conference of the OUP & Centre for the New OED and Text Research* University of Waterloo Centre for the New OED p. 129-136.

PRÓSZÉKY G. - TIHANYI L.: A Fast Morphological Analyzer for Lemmatizing Corpora of Agglutinative Languages.

A Fast Morphological Analyzer for Lemmatizing Corpora of Agglutinative Languages

GÁBOR PRÓSZÉKY — LÁSZLÓ TIHANYI

Abstract

HUMOR, a general-purpose morphological processor is designed to perform both analysis and synthesis. Its first complete implementation (that handles hundreds of millions of Hungarian word forms[1]) and its first application — lemmatizing a large corpus used for building *Dictionary of Hungarian Based on Historical Principles* at the Lexicographic Department of the Research Institute for Linguistics — are introduced in this article. Several by-products of the system, like a fault-tolerant version of the analyzer, morphological support components to hyphenation and synonym-handling, are also shown.

1. Introduction

In contrast to major Indo-European languages, such as English, a single lexical entry of agglutinative languages can occur in several hundreds or thousands of different shapes depending on the form of the suffix combination that actually follows it. Examples of Hungarian, the first language our system has been implemented for, illustrate this:

programjainkban:	*program+ja+i+nk+ban*	*[in our programs]*
programotokról:	*program+otok+ról*	*[about your program]*

[1] This number is very difficult to define because of the formal generation of non-existing (meaningless) but theoretically acceptable forms.

Chunking suffixes does not help us to identify the stem because phonologically motivated rules frequently change the form of the stem, too (the common part of the two word-forms here is *ke* only):

kehely		*[drinking cup]*
kelyhet	*kelyh+et*	*[drinking cup + ACC]*

In spite of the first implementation of the system that concentrates on Hungarian (or, more exactly, written Hungarian), other agglutinating or morphologically simpler languages also can be treated by it. The linguistic method we use is based on surface strings. Unification[2] is the only applied operation, so neither transformations nor "human-unreadable" lexical forms (as in two-level systems, Koskenniemi 1983) are used.

The system called HUMOR[3] is fully implemented for Hungarian. Its stem dictionary contains 80.000 stems and covers all (cca. 70.000) lexemes of the *Explanatory Concise Dictionary of the Hungarian Language.* Suffix dictionaries contain all the inflectional suffixes and the productive derivational morphemes of present-day Hungarian. HUMOR implementation makes analysis and generation possible, morphologically supported hyphenation, existing spell-checkers and synonym dictionaries also rely on it. The main reasons why we did not use an existing product — practically one of the implementations of two-level morphology (Koskenniemi 1983) — are the followings:

- QUICK AND EASY MODIFICATION POSSIBILITIES: we wanted an easy-to-modify system, since compilation time of two-level systems are enormously long,
- EXTENSIVITY: we wanted to handle special phenomena (context-sensitive morphemes) internally since it cannot be done internally in two-level systems, only by off-line modules,
- NEW LEXICONS: we needed easy adaptation possibilities of new lexicons,
- SPEED: we wanted a very fast and configurable system the speed of which did not change when large dictionaries were adapted to it,
- INTEGRATION INTO NUMEROUS APPLICATIONS: we wanted to integrate the kernel module of the system into several applications (lemmatization, hyphenation, thesaurus etc.) therefore we wanted to maintain easy modification possibilities of the source program,
- COMMERCIAL APPLICATIONS: we also wanted to produce usable commercial linguistic software tools integrated into word-processors.

---

2 It is the normal feature unification and not the string unification of Calder (1989).

3 It stands for: high-speed unification morphology

2. Unification morphology

Unification-based morphology relies on the fact that the behavior of stems or stem allomorphs in word-forms can be morphologically and morpho-phonologically described by the values of the relevant features of the affixes the stem in question precedes. A morphological analyzer based on this principle can cope with problems of agglutinative and other (highly) inflectional languages very effectively. The first language that HUMOR has been applied to is Hungarian.

Segmentation of a word-form in HUMOR is based on lexical patterns, that is, typical sequences of separate suffix morphemes, e.g. the following two morphemes are generally accepted as suffix sequences analyzed as a whole:

+je+i+tek:	PER + PL-POSS + 2nd + PL	*-jeitek*
-é-i:	POSS + PL-POSS	*-éi*

Following the above examples, other sequences can also be constructed, e.g.

-é-i-tek:	POSS + PL-POSS + 2nd + PL	*-éitek*

These patterns are generated in an earlier phase of development by hand, or by an off-line morphological generator which need not to be as fast as the run-time module of HUMOR (for details, see later: 3.2, 3.3). Running this generator can be considered the learning phase of the algorithm. The generated stem variants and suffix combinations are stored in an internal lexicon structure that guarantees very fast searching. The full algorithm simulates a hypothesis according to which most segments of word-forms in agglutinative languages are handled as "Gestalts", instead of parsing them on-line (see 3.3).

Features used for checking appropriate properties of stems and suffixes are orthogonal properties of morpho-graphemic behavior (see: 3.4). Stem allomorphs are classified according to paradigm types they belong to. Suffixes are classified according to their participation in the paradigms (see: 3.1). Checking appropriateness is based on unification of the adequate features of stems and suffixes (see: 3.4).

3. Organization of lexicons

Every morphological system consists of two parts: lexicons of stems and suffixes with adequate morphological and morpho-phonological information. The algorithmic part that knows all the rules of Hungarian morphology and morpho-phonology and is able to apply them very fast. Lexicon structure plays a central role in HUMOR: the algorithmic part consists of implementation unification operation only (see 3.4.).

3.1 Paradigm groups and paradigms

Two main paradigm groups are distinguished, as usual: nominal and verbal. Concrete nominal and verbal paradigms are defined by the sets of affixes the stems belonging to the paradigm can be followed by directly.

Both paradigm groups can be represented as nets, where the nodes are labeled with the concrete paradigms. (Partial) ordering of the set of paradigms is defined by the degree and sort of defectivity. It means, for example, in linguistic terms, that adjectives and numerals have all the attributes that nouns generally have, but might be followed by suffixes which never occur after a noun. Therefore the degree of defectivity of a common noun is greater than of either adjectives or numerals, thus a slice of the partially ordered structure can be shown as follows:

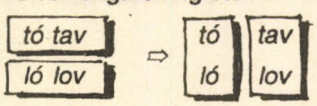
{NUMERALS,ADJECTIVES} > NOUNS > {PERSPRONS, POSTPOSITIONS, ADVERBS}

Stems that cannot be followed by any suffixes (ARTICLES, CONJUNCTIONS, VERBALPREFIXES, etc.) belong to the most defective paradigms, therefore they are located at the lowest level in the partial hierarchy.

3.2 Base forms vs. allomorphs

According to Karlsson (1986) there are several ways in which the lexical forms of words may be construed: full listing, minimal listing, methods with unique lexical forms and methods with phonologically distinct stem variants. Variants of full listing do not need rules at all, but are implausible for agglutinative languages. Minimal listings need a quite large rule system in case of highly inflectional languages, although their lexicons are relatively small. In methods based on unique lexical forms allowing diacritics and morpho-phonemes (Koskenniemi 1983, Abondolo 1988) paradigms are represented by a single base form as in case of minimal listing methods, but the number of rules is small. Finally, the representation utilized here regards phonologically distinct bound variants of a base-form as separate stems.[4] Two variants of the method are known: the one using technical stems and another using real allomorphs. The former was used in the TEXFIN system of Karttunen (1981), the latter was applied by Karlsson (1986) and this is the method we have chosen in the system to be introduced here.

Our lexicons contain stem allomorphs (generated by the learning phase) instead of single stems. Relations among allomorphs of the same base-form are mainly important for syntax/semantics and the end-user. On-line morphological parsing need not be directly interested in the derivation of allomorphs from their base-forms, like vowel-deletion, vowel lengthening etc.:



4 Concrete two-level (and some other) descriptions apply similar methods in order to cope with morphotactic problems which cannot be treated phonologically in an elegant way.

While learning and storing lexical data, however, the above relations have to be known. In case of a natural language processing system, learning can be replaced by inputting linguistic knowledge into the system. Storing lexical information must be as redundancy-free as possible. Thus, in the above example, relations *tó/tav* and *ló/lov* have been learned earlier by the system, although this knowledge might not be used run-time. The same is true in general: avoiding redundancy is important, however, not by increasing the complexity of words: when the learning phase is over, no previously used algorithmic learning steps need to be repeated, but practically easy-to-use "Gestalts" are applied instead.

3.3 Affixes vs. affix arrays

"A psycholinguistic argument for treating (some) ending sequences as wholes comes from the observation that children acquiring inflectional languages seldom make errors involving the order of morphemes in a word." (Bybee 1985, p. 114) "The endings and entries are often listed as wholes, especially in close-knit combinations.[5] Such combinations are often subject to bi-directional dependencies that are hard to capture otherwise" (Karlsson 1986, p. 106).

Researchers of Hungarian morphology do the same when trying to analyze relations between definite and indefinite verbal suffixes, because this relation is relevant from a diachronic point of view only[6]:

látunk:	*lát+unk < lát+u+m+k*	*[we see]*
látjuk:	*lát+juk < lát+?+k*	*[we see it]*

In spite of the surface transparency, some orthographically motivated phenomena of the Hungarian nominal paradigm have to be treated in the same manner. E.g. no three copies of the same consonant can occur in a row:

sakkal:	*sakk+al < sakk+kal*	*[with chess]*

Both verbal and nominal stem allomorphs can be characterized by the set of suffix allomorphs that can follow them. Suffixes — according to the principles sketched above — are either single suffixes or suffix-complexes. When describing the behavior of stems, all the suffix-combinations beginning with the same morpheme are considered equal, because the only relevant information comes from the suffix that immediately follows the stem:

kalapja:	*kalap+ja*	*[his hat]*
kalapjai:	*kalap+ja+i*	*[his hats]*
kalapjairól:	*kalap+ja+i+ról*	*[about his hats]*

We consider linking vowels occurring between stem and the first suffix belonging to the latter, since less stem allomorphs might be used this way:

---

5 A good example for this are number and person combinations of Hungarian definite conjugation (see the following example).

6 For another example, see the introductory Section 2.

Stem ending vowel the critical vowel	Suffix beginning with the critical vowel	
① *ház=*	① *ház =*	*[house]*
② *háza=m*	① *ház =am*	*[my house]*
③ *házu=nk*	① *ház =unk*	*[our house]*

One of the most important advantages of this method is that no traditional categories of suffix classification need to be used; nothing but the relevant factors have to be defined. The only thing we must know is which elements can occur in which positions following the stem.

3.4 Morphotactics and morpho-phonology

Attributes on the basis of which concatenation of a stem allomorph and a suffix allomorph can be considered regular or irregular are classified in terms of the following two factors:

- continuation classes[7] defined by the paradigm descriptions, and
- classes of surface allomorphs (as a cross-classification of the above paradigms).

Every affix array is represented by its starting affix, because there is an equivalence relation on the set of affix arrays. An abstract name is given to each equivalence class and each paradigm, that is, each existing set of equivalence classes has an abstract name, e.g. {PERS, POSS, PL, CAS} is the full noun paradigm. Its subsets are {PL}, {PERS, POSS, CAS} etc.

For example, the stem *ház [house]* has the paradigm that can be described by the set {PERS, POSS, PL, CAS}. *atok* is a suffix belonging to CAS, thus word form *ház+atok is* morphotactically licensed.

Features (morpho-phonological properties) are used to characterize both stem and suffix allomorphs. A list of *feature=value* pairs in brackets shows the morphological structure of morphemes:

ház: [Vow=Back, Rnd=Round, Acc=V, Pl=V,Pers3=0,PersPl2= V,...]
atok: [Vow=Back, Rnd=Round, PersPl2=V]

Since features of the above suffix allomorph form a subset of features of the stem allomorph having the same values — that is, their unification is succesful — the word-form *ház+atok* is also morpho-phonologically licensed.

Generative production of the allomorph set and providing lexemes with codes based on the above classifications were our first tasks when implementing HUMOR. It could be done half-automatically with the help of programs written directly for these purposes.

The feature-based method's main advantage is that possible paradigm and morpho-phonological types need not be defined previously, but classification criteria have to be clarified only. Since the quantity of these criteria is around a

---

[7] Similar to the two-level descriptions' continuation classes (Koskenniemi 1983).

few dozens (in case of a language with rather complicated morphology[8]), the number of theoretically possible paradigm classes is several millions or more. In case of Hungarian we have chosen about ten orthogonal properties which define a thousand possible classes, but, in fact, a large subset of these hypothetical classes does not occur in Hungarian.

It is not worth trying to define a hierarchical ordering of features in HUMOR systems because they are orthogonal, or in other words, independent features. There is also no ordering among the elements of the set of the existing feature configurations, but the distance of any two elements of this set can be defined on the basis of the number of features having different values. Clustering also can be done, but neither this nor the distance calculations influence the description — it might be used, at most, for making traditional definitions of morpheme classes more exact than they are usually generally defined.

3.5 Derivational vs. non-terminal suffixes

The application of derivational suffixes in morphological recognition and synthesis programs always raises the question whether there is a strict border between "real" derivational suffixes which belong to the kernel paradigm (e.g. participles) and inflectional suffixes, or not. We do not want to swell the pros and cons[9], but suggest another distinction between derivational(-like) and inflectional(-like) suffixes. The idea is the following: non-terminal suffixes are elements of morpheme sequences that can (but need not) be followed by other — so-called terminal — suffixes which cannot be followed by anything:

játszhat	*játsz+hat*	*[(he/she) can play]*
játszunk	*játsz+unk*	*[we play]*
játszhatunk	*játsz+hat+unk*	*[we can play]*
**játszunkhat*	**játsz+unk+hat*	

That is: *hat* is non-terminal because it can be followed by other suffixes, *unk* is terminal since it is always the last element of the suffix sequence.[10]

3.6 Compounding

Compounds can be lexicalized (and therefore contained by our lexicon) or can be built productively by the algorithmic part of the system. Productive compounding rules in the Hungarian version of HUMOR include:

- some unmarked noun-noun sequences,
- certain unmarked adjective-noun sequences,
- regular numeral-numeral sequences,
- verbal prefix-verb pairs,
- superlative adjectives.

---

8 Hungarian is a good choice from this respect as the first implementation.

9 MMNyr (1961) characterizes all the derivational suffixes we use as productive.

10 We remark here that all the non-terminal suffixes of our system are characterized as productive by MMNyR.

4. Applications of HUMOR

In order to get a realistic picture of HUMOR all the application areas where the kernel algorithm has been used are introduced in this chapter. These are: lemmatization, recognition of unknown variants of known words, supporting hyphenation algorithms, synthesis, and the development of an inflectional thesaurus system.

4.1 Lemmatization

Lexicographers would like to see concordance lists ordered according to lexemes. In case of agglutinative languages a morphological analyzer is the only solution to offer lexemes instead of different word-forms. The lemmatizer's role is played by HUMOR in the "Hungarian Historical Dictionary" project of the Research Institute for Linguistics, where approximately 5 million running words of 20th century texts are and will be analyzed. Lemmas are parts of speech and special information about inflections. Here is an example:

INPUT: *átirányíthatóságaitokéira*
OUTPUT: *át[IK]·irányít[IGE]·hat[HAT]·ó[MIF]·ság[COL]·aitok[PSt2i]·éi[POS]·ra[SUB]*

Some word-forms have more than one segmentations. This fact is marked in the output file. An example for multiple outputs is given below:

INPUT:	*mentek*	
OUTPUT:	*ment[MN]·ek[PL]*	*[the saved ones]*
	ment[IGE]·ek[e1]	*[I save]*
	megy[IGE]≈men·tek[Mt3]	*[they went]*
	megy[IGE]≈men·tek[t2]	*[you -PL went]*
	megy[IGE]≈men·t[MIB]·ek[PL]	*[ones that had gone]*

Earlier (16th–19th century) materials can also be processed by the same method, but different lexical entries, other features, that is, different morphological grammars would be written for those purposes. Another problem when trying to apply spelling correction programs (see: 3.2) to machine-readable versions of early printed texts is the inconsequent spelling used in them which makes application of a system relying on consistent orthography very difficult. An example for the lemmatization of a rather long text with corrections of 'errors' is given in the Appendix.

4.2 Recognition of unknown variants of known words

A list of all possible forms of a Hungarian dictionary would contain hundreds of millions of word forms (see footnote 1), thus the well-known word-list based methods for spell-checking cannot be applied.[11] Fortunately, a fast

[11] For comparison only: the size of the average English dictionaries used in spell-checkers would contain less than 100 Hungarian words' all possible forms with the same data compression method.

morphological analyzer helps to solve this problem of spell-checking in agglutinative languages. Two possible applications of spell-checking mode of HUMOR have been worked out:

- identifier of unknown variants of known words, that is, morphological analyzer is augmented by a guessing component
- "real" spell-check mode, when it is satisfactory if the system finds the first possible segmentation of a word form, thus giving a binary answer whether the word is acceptable or not. If no other output information is needed, "real" spell-checking is, of course, faster than analysis.

Suggestions for corrections are mainly based on linguistic and orthographic phenomena but corrections of typographical errors must also be supported:

- phonographic errors
  - bad spelling of assimilations (e.g. **haggy ⇨ hagyj*)
  - bad spelling of short/long consonants (e.g. *elô ⇨ ellô*)
  - traditional orthography vs. pronunciation (*j ⇨ ly, c ⇨ cz, y ⇨ i* etc.)
- orthographic errors
  - one word/two words errors (perhaps the most frequent error in Hungarian, e.g. **jobbkéz ⇨ jobb kéz*)
  - abbreviations with/without ending dots (*m, mp, Ft,* but: *stb.*)
  - wrong affixation of numbers, signs and abbreviations
  - capitalization errors (proper names if not followed by derivational suffix *i*)
  - violations of the so-called 6-3 rule (e.g. *adatbázisgenerátor ⇨ adatbázis-generátor*)
  - wrong hyphenation (see: Section 4.3)
- language-dependent typographic errors
  - short/long vowel error (*i ⇨ í, o ⇨ ó, u ⇨ ú* etc.)
  - mis-typing caused by differences between English and Hungarian keyboard layouts (*y ⇨ z*, positions of special Hungarian vowels etc.)
- language-independent typographic errors
  - deletion (e.g. *kelett ⇨ kelet*)
  - insertion (e.g. *kelett ⇨ kellett*)
  - substitution (e.g. *keleq ⇨ kelet*)
  - inversion (e.g. *kelett ⇨ keltet*)

It is important to note here that the set of the above phenomena can be modified with the help of a control program, therefore more traditional orthography which is typical in case of tagging literary texts of 19th or early 20th century can also be treated. It is also possible to change the whole configuration in order to be able to analyze texts of other input type, e.g. scanned texts.

4.3 Supporting hyphenation algorithms

Certain morpheme boundaries can override the main and simple hyphenation rules. Hence, morphological analysis is needed in order to make perfect hyphenation in Hungarian. Compounds and words beginning with verbal or superlative prefixes need a generative process to identify their boundaries. It is much better than an algorithm using an always unsatisfactory finite list of so-called "exceptions". Some examples are given below:

Word	Hyphenation	Type and meaning
Compound		
filétek	*fi-lé-tek*	not compound (*filé*+Suffix): 'your fillet'
csalétek	*csal-étek*	compound (*csal+étek*): 'allurement'
Two possible segmentations: simple and compound		
gépelem	*gé-pe-lem*	simple (*gépel*+Suffix): 'I type it'
	gép-elem	compound (*gép+elem*): 'machine part'
Long digraph vs. morpheme boundary in compound		
karosszéria	*ka-rosz-szé-ria*	not compound: 'car-body'
karosszék	*ka-ros-szék*	compound (karos+szék): 'armchair'
Superlative prefix		
legelőre	*le-ge-lő-re*	no prefix: 'to grazing lands'
	leg-elő-re	prefix (leg+előre): 'to the first place'
Verbal prefix		
átall	*átall*	no prefix: 'hate (to do sg)'
átáll	*át-áll*	prefix (át+áll): 'go over (to the other side)'
Simple word vs. prefixed word		
megint	*me-gint*	simple: 'again'
	meg-int	prefixed (meg+int): 'he/she warns sy'

4.4 Morphological synthesis

Morphological synthesis is the composition of a surface string from a base form and some additional information encoded into separate affixes. Since formalized morphological information is difficult to memorize, the affixation process is given a stem and a sample word form bearing the same tokens of affixes the base form in question must be provided with. For example,

SAMPLE AFFIXED FORM:	*gyerekeimnek*	*[to my children]*
MORPHOLOGICAL ANALYSIS:	*gyerek +eim+nek*	
INPUT STEM'S BASE-FORM:	*nagyapa*	*[grandfather]*
MORPHOLOGICAL SYNTHESIS:	*nagyapá +im+nak*	
OUTPUT:	*nagyapáimnak*	*[to my grandfathers]*

4.5 Inflectional thesaurus for word-processors

Thesaurus — as synonym dictionaries are called in word processor environments — is a well-known linguistic software tool. In inflectional languages, the simple replacement of a running word-form with a base-form is not sufficient. In order to cope with the problems of the substitution of one Hungarian word-form with another, an inflectional thesaurus had to be developed. The kernel thesaurus options are the usual; however, its input is not the original word-form of the text but its lexical base form, and its output is not simply the chosen synonym. Rather, this synonym is provided with the same morphological information as the original input word was. The synthesis module as introduced above generates the appropriate inflected form (applying rules for vowel harmonies, assimilations etc.). E.g.

WORD-FORM TO BE REPLACED:	*kupáimra*		*[to my drinking cup]*
MORPHOLOGICAL ANALYSIS:	*kupá*	*+im+ra*	
BASE-FORM OF ITS STEM:	*kupa*		*[drinking cup]*
BASE-FORM OF SYNONYM CHOSEN:	*kehely*		*[drinking cup]*
MORPHOLOGICAL SYNTHESIS:	*kelyh*	*+eim+re*	
REPLACING WORD-FORM:	*kelyheimre*		*[to my drinking cup]*

5. Implementation and evaluation

HUMOR is fully implemented for Hungarian. Its stem dictionary contains 80.000 stems which covers all (approx. 70.000) lexemes of the *Concise Explanatory Dictionary of the Hungarian Language*. Suffix dictionaries contain all the inflectional suffixes and the productive derivational morphemes of present-day Hungarian. With the help of these dictionaries HUMOR is able to analyze and/or generate around 1.000.000.000 well-formed Hungarian word-forms. HUMOR implementation makes not only analysis and generation possible but numerous applications, like morphologically supported hyphenation, spell-checkers, spelling correctors and synonym dictionaries also rely on it.

The whole system is written in programming language C. It uses around 100 kilobytes of RAM. The dictionary of 80.000 entries requires around 600 kilobytes hard disk space for the spell-checking version and an additional 200 kilobytes for the analyzer version. These latter data containing lemmas and other human-readable output information have not been compressed yet. Its speed is 35 ms/word form on a 20 MHz PS/2.[12]

Several stand-alone versions of the program and applications based on it run under MS-DOS and MS-Windows on PCs and a Macintosh version is under testing.

---

12 Tests have been made after running the disk re-organization program SpeeDisk.

6. Development plans

Development plans can be summarized by the following keywords: improvement, new languages, new application areas and better user interface.

- Improvement of the system means correction of the present errors, mistypings and other disorders of the stem and suffix lexicons.
- Besides the Hungarian version of HUMOR, an English morphological grammar that is complete with respect to inflection has also been implemented, but its stem dictionary, at least for the time being, serves demonstration purposes only. Morphological description of Polish for HUMOR is under development, a large subgrammar works already. Development of Latin and Turkish versions have also begun.
- The most important new application area where the HUMOR method will be used is syntax. This means that the idea beyond morphological parsing might be applied (with little modifications) on the syntactic level, too. Elaboration of the details has already begun. An eventual by-product of this development may be a grammar-checker.
- A better user interface is also needed because, for the time being, all the functions of the system cannot be studied at the same time.

7. Conclusion

HUMOR systems analyze and/or build arbitrarily complex surface forms of agglutinative (and, of course, other, morphologically simpler) languages. HUMOR systems need little memory space and run fast even on PC's, thus it can be used for lemmatizing large files, spell-checking and correcting texts of agglutinative and other highly inflectional languages. The reason of its reasonable speed and simple architecture is that as many relevant morphological and morpho-phonological phenomena are analyzed off-line (in an earlier phase) as possible, and for on-line parsing only one very fast operation is used: non-destructive feature unification.

8. References

Abondolo, D. M. (1988)
Hungarian Inflectional Morphology. Akadémiai, Budapest.

Bybee, J. L. (1985)
Morphology. A Study of the Relation between Meaning and Form. Benjamins, Amsterdam.

Calder, J. (1989)
Paradigmatic Morphology. *Proceedings of 4th Conference of EACL 89*: 58-65.

Jäppinen, H. and Ylilammi, M. (1986)
Associative Model of Morphological Analysis: An Empirical Inquiry. *Computational Linguistics* 12(4): 257-252.

Karlsson, F. (1986)
A Paradigm-based Morphological Analyzer. *Papers from the Fifth Scandinavian Conference of Computational Linguistics,* Helsinki: 95-112.

Karttunen, L., Root, R. and Uszkoreit, H (1981)
Morphological Analysis of Finnish by Computer. *Proceedings of the 71st Annual Meeting of the SASS.* Albuquerque, New Mexico.

Koskenniemi, K. (1983)
Two-level Morphology: A General Computational Model for Word-form Recognition and Production. Univ. of Helsinki, Dept. of Gen. Ling., Publications No.11.

MMNyr (1961)
System of Present-day Hungarian. Descriptive Grammar. [in Hungarian] Akadémiai, Budapest.

Prószéky, G., Kiss, Z. and Tóth, L. (1982)
Morphological and Morphonological Analysis of Hungarian Word-forms by Computer. *Computational Linguistics and Computer Languages.* Vol. XV. 195–228.

Prószéky, G. (1989)
Computational Linguistics [in Hungarian]. Számalk, Budapest,

Slocum, J. (1988)
Morphological Processing in the Nabu System. *Proceedings of the 2nd Applied Natural Language Processing:* 228-234.

Appendix

Áprily Lajos: Októberi séta

ORIGINAL INPUT TEXT	TEXT LEMMATIZED BY HUMOR
Ez itt a hervadás tündér-világa.	Ez[NM] itt[HA] a[DET] hervadás[FN] tündér[FN]-világ[FN]·a[PSe3].
Akartál látni szép halált velem?	Akar[IGE]·tál[Me2] lát[IGE]·ni[INF] szép[MN] halál[FN]·t[ACC] vele[HA]·m[PSe1]?
A Bükkös-erdő bús elégiája	A[DET] Bükk[FN]·ös[SKEP]-erdő[FN] bús[MN] elégia[FN]≅elégiá·ja[PSe3]
szép mint a halál és a szerelem.	szép[MN] mint[KOT] a[DET] halál[FN] és[KOT] a[DET] szerelem[FN].
Fától fához remegve száll a sóhaj,	fa[FN]≅Fá·tól[ABL] fa[FN]≅fá·hoz[ALL] remeg[IGE]·ve[HIN] száll[IGE] a[DET] sóhaj[FN],
közöttük láthatatlan kéz kaszál.	között[HA]·ük[PSt3] láthatatlan[MN] kéz[FN] kaszál[IGE].
Az ágakról a fölrebbent rigóraj	Az[DET] ág[FN]·ak[PL]·ról[DEL] a[DET] föl[IK]·rebben[IGE]·t[MIB] rigó[FN]·raj[FN]
tengődni még a holt irtásba száll.	tengődik[IGE]≅tengőd·ni[INF] még[HA] a[DET] holt[MN] irtás[FN]·ba[ILL] száll[IGE].
Lombját a galy, nézd, mily kimélve ejti,	Lomb[FN]·já[PSe3]·t[ACC] a[DET] galy^gally[FN], néz[IGE]·d[TPe2], mily[NM] kimélve^kímél[IGE]·ve[HIN] ejt[IGE]·i[Te3],
holnap szél indul, döntő támadás,	holnap[HA] szél[FN] indul[IGE], döntő[MN] támadás[FN],
holnaputánra minden elfelejti,	holnapután[HA]·ra[SUB] minden[NM] el[IK]·felejt[IGE]·i[Te3],
milyen volt itt a végső lázadás.	milyen[NM] van[IGE]≅vol·t[Me3] itt[HA] a[DET] végső[MN] lázadás[FN].
Mint gyertya-csonkok roppant ravatalnál,	Mint[KOT] gyertya[FN]-csonk[FN]·ok[PL] roppant[MN] ravatal[FN]·nál[ADE],
tönkök merednek dúltan szerteszét	tönk[FN]·ök[PL] mered[IGE]·nek[t3] dúlt[MN]·an[ESS] szerteszét[HA]
s a nyár, ez a kilobbant forradalmár,	s[KOT] a[DET] nyár[FN], ez[NM] a[DET] ki[IK]·lobban[IGE]·t[MIB] forradalmár[FN],
vérpadra hajtja szőke szép fejét.	vérpad[FN]·ra[SUB] hajt[IGE]·ja[Te3] szőke[MN] szép[MN] fej[FN]·é[PSe3]·t[ACC].

Notes:

Each morpheme is followed by its lemma in brackets. A dot between two annotated morphemes mean that they occurred in the same word-form. The sign ^ binds two morphemes: the first is the original form as it occurred in the text, the second is its corrected version. It refers to fault-tolerant behavior of HUMOR, mentioned earlier. ≅ characters also link two morphemes to each other: the first morpheme is the base-form of the second one which originally occurred in the text. Lemmas are written after the base-forms. The list of annotations with their meanings are listed below: ABL (ablative), ACC (accusative), ADE (adessive), ALL (allative), DEL (delative), DET (determiner), ESS (essive), FN (noun), HA (adverb), HIN (gerund), IGE (verb), IK (verbal prefix), ILL (illative), INF (infinitive), KOT (conjunctive), Me2 (past, sg, 2nd pers), Me3 (past, sg, 3rd pers), MIB (past participle), NM (pronoun), PL (plural), PSe1 (pers. sfx, sg 1st pers), PSt3 (pers. sfx, sg, 3rd pers), SKEP (deriv. suffix *-s*), SUB (subessive), t3 (plural, 3rd pers), Te3 (defin., sg, 3rd pers.), TPe2 (defin., imp., sg, 2nd pers).

Looking for syntactic patterns in texts

EMMANUELE ROCHE

1 Summary

We here show that a syntactic dictionary of verbs (i.e. a dictionary of the argument structure of the verbs) can be used to build a pattern matching system for French texts, that is, a complex syntactic pattern such as a verb with one of its specific complement can be an argument for the matching function (we focus on the preposition that links the verb and its complement). The formalism used is that of Finite State Automata (FSA). We present here a FSA syntactic dictionary together with experiments on corpora . Experiments are performed on French texts with large scale dictionaries: a 7,000 verbs syntactic dictionary, a morphological dictionary of 600,000 simple inflected forms and a morphological dictionary of 150,000 inflected compound nouns.

2. Introduction

Suppose we want to recognize in texts complex patterns such as a verb followed by a possibly prepositional complement, for instance one of the sequences *buys this* or *gives to Peter*. Simple morphological analysis of the text leads to irrelevant matchings like in the following examples:

Consider the following sentence

*(1) He read the **charge against them**,*

The morphological ambiguity of *charge* (noun or verb) leads to a local analysis where *against them* is the complement of the verb *charge*. The problem found in (1) can sometime be solved by local disambiguation rules. On the other hand consider the sentence:

*(2) He **eats on the floor**,*

on the floor is an adverbial complement, not specific of the verb *eat*. But this complement can be said not relevant to our question (we search for specific complement) only if one has an exhaustive description of the argument structure of this verb, namely a description that distinguishes specific from non specific complements. For instance one should know that *to eat* cannot enter the basic structure N_0 *V on* N_1 whereas *to decide* in *He decided on a new floor* has this basic structure. Hence the sequence *eats on the floor* does not answer our question. We thus need to know for matched verbs whether they have specific complements and, if yes, what are the prepositions that introduce them.

We have at our disposal a complete description of the argument structure of the verbs (syntactic tables built at LADL), we have translated it into Finite State Automata formalism (F.S.A.). This formalism appeared to be very convenient from a linguistic point of view and it leads to efficient computer implementation.

We will briefly describe how the syntactic properties of a verb can be expressed within FSA graphs which will constitute a syntactic dictionary.

3. Overview of the linguistic background

This syntactic description of the argument structure of verbs, that is of the elementary sentences in which they can take place, has been done for French at the LADL (Paris 7) over the past twenty years. This effort led to a 12,000 verb syntactic description called a lexicon-grammar. It is subcategorized into syntactic tables that describe verbs that share defining properties. The following figure is a sample of one of these tables.

Sujet					Adjectif				Comp. direct									
N_{hum}	N_{nr}	= fait Qu P	V^1 Ω		V "concret"	N_0 V	= être	= avoir	= être - être sur	N_{hum}	N_{hum}	= fait Qu P	N_1 = V de ce Qu P	N_1 = V numéro de N_{hum} de ce Qu P	N_1 est V par de ce Qu P	[passif par]	[passif de]	N_0 V N_1 contre N_{hum}
+	+	+	+	abasourdir	–	+	+	–	–	–	+	–	–	–	+	+	–	–
+	+	+	+	abattre	+	+	–	–	–	+	+	–	+	–	+	+	–	–
+	+	+	+	abîmer	+	+	–	–	–	+	+	–	–	–	–	+	–	–
+	+	+	+	abrutir	–	+	+	–	–	+	–	–	–	–	–	+	–	–
–	+	+	+	absorber	+	+	+	–	–	+	–	–	–	–	–	+	–	–
+	+	+	+	abuser	–	+	–	–	–	+	–	–	–	–	+	+	–	–

Figure 1 Table 4. QuP V Nhum (see Maurice Gross 1975).

Each row of this binary matrix represents an entry and each column represents a syntactic property which stands for a simple sentence structure into which the verb may enter ("+" mark) or not ("-" mark).

4. Description of the syntactic dictionary

Consider, for example, the French verb *abasourdir* (*to astonish*). The entry in our dictionary is the canonical form , that is the word *abasourdir* in the infinitive form, and the syntactic information is given in the FSA figure 2. This FSA describes the set of accepted sentence structures for this verb. For instance, the path *Qu P V Nhum* is allowed by this graph, this means that the structure *Sentence subject (Qu P) followed by a verb (V) followed by a human object (Nhum)* is allowed for this verb (for instance in *Que Pierre fasse cela abasourdit Paul* (*That Peter did this astonishes Paul*)). The verb is syntactically unambiguous but in the case of a verb like *voler* (meaning either *to fly* or *to steal*) the entry would point to two FSAs, that is one for each reading.

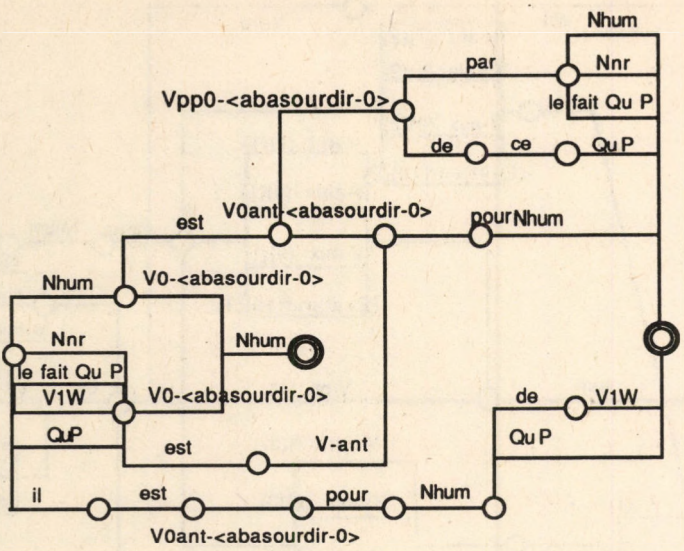


Figure 2: Entry of *abasourdir* in the syntactic dictionary of verbs[2]

5. Building a full syntactic dictionary

Building a complete dictionary out of[3] the set of syntactic tables (like the one figure 1) is not straightforward. In fact, it is not possible to generate directly the FSA out of an entry description because of the lack of formal specification of the tables specification. It is also not possible to design each FSA by hand, it would take too long. The solution which was adopted is the following:

1. One designs by hand a reference FSA for each syntactic class (i.e. table). This FSA stands for an abstract verb that would have all the properties of the class. Then, each property (i.e. column of the table) points to a set of transition in this FSA.

2. The FSA of a particular verb is then the reference FSA minus the transition pointed to by the properties negatively marked.

The FSA of *abasourdir* was thus generated out of the reference FSA figure 3. For instance, since *abasourdir* can not be derived into an adjective of shape *-ble*, this is specified in the 8th column, all the transitions marked by (8) in figure 3 are to be deleted.

Moreover, storing all the FSA of all verbs would need a huge amount of storage. This, in turn, would slow the access to the information. On the other hand, this solution requires the storing of a small set of FSA (below 50) and the storage of the properties of the verbs as they were in the table (which can be done in a bit field).

---

[2]The abbreviations respectively have the following meanings: Nhum: human substantive, Nnr: semantically unrestricted subtantive, QuP: That sentence, V1W: infinitive phrase the subject of which is the complement N_1, Vpp: past participle, V-ant: present participle. The reading number of the actual verb is given with him: abasourdir-0 stands for "the first reading (reading number 0) of abasourdir".

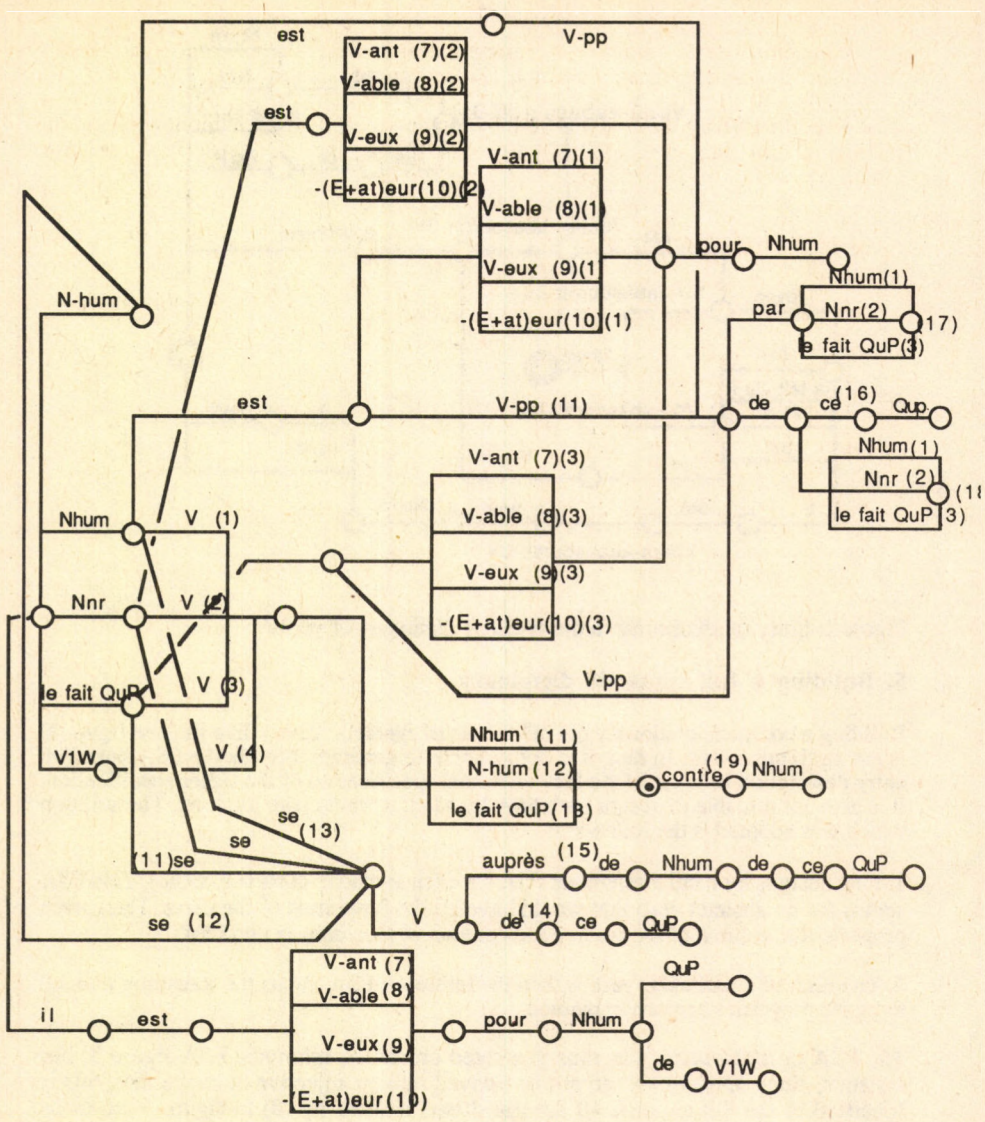


Figure 3

6. Description of the experiment.

Suppose that, like in examples already discussed, we want to find in a French text every occurrence of the sequence: *Verb followed by one of its complement.* We could proceed in the following way:

1. According to the morphological dictionary we select all the sequences *Verb*

(E+Preposition) Article Noun[4]. This grammatical analysis also provides syntactically irrelevant structures like that of example (2) (*eats on the floor*).

2. We get the FSA of the detected verb by looking up the syntactic dictionary and the mentioned structures are checked against this FSA. The correct sequences are those which are matched in this process.

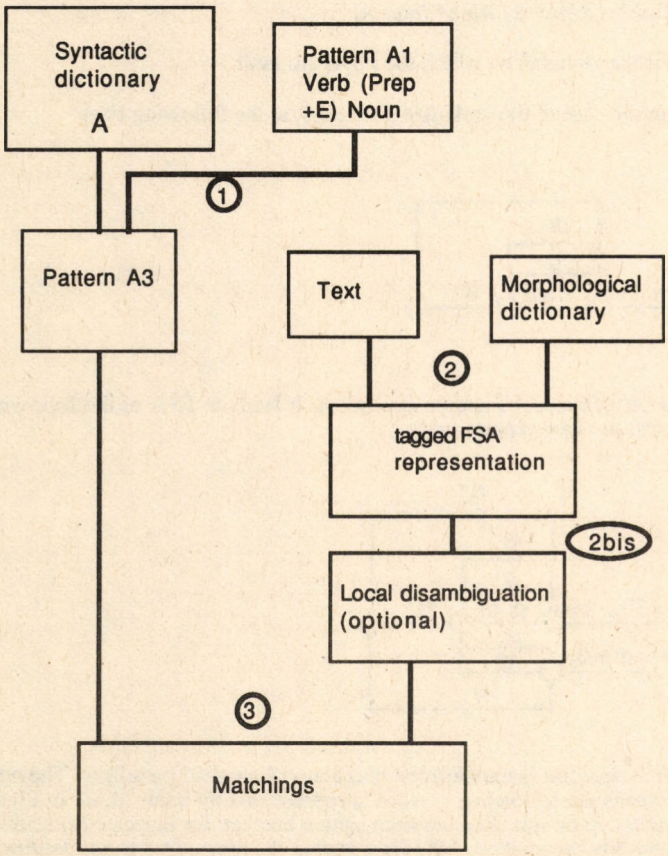


Figure 4

However, if the principle is roughly this one, we take advantage of the power of the formal operations on FSA to improve the efficiency of the matching operation in the following way:

A3 is the pattern that, in our example, represents all the *verb preposition* combinations. It is a FSA that, for instance, contains the *abasourdir E Noun* sequence. Let us first remark that since the pattern A3 is independent of the text, their is no reason to compute it dynamically. In fact the computation of A3 (which is the circle number 1 of figure 4)

---

[4]E stands for the empty string. Of course we have oversimplified the structure of the noun phrase.

is done in the following way:

1. For each verb of the dictionary, its FSA A_i is computed (as described in section 5)(i takes its values between 1 and the number of verb in the syntactic dictionary) .

2. For each of these FSAs one computes its intersection A'_i with the pattern A_1 *(Verb preposition noun sequence)*. More precisely, the result A'_i is defined by

$Alph^* A'_i Alph^* = Alph^* A_1 Alph^*$ *inter* A_i

where *Alph* is the alphabet on which the FSAs are built

This leads, in the case of the verb *dire-1* (*to say*), to the following FSA:

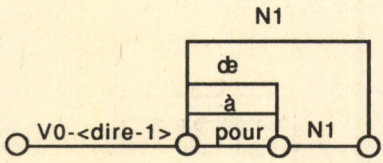


Figure 5

3 The union *A"* of these A'_i is then computed. It leads to FSA of the following shape (only two verbs are here represented):

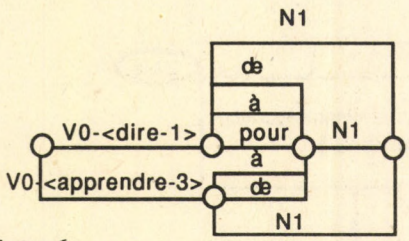


Figure 6

4. Each of the transition is transformed into a morphological paradigm. The notation of these paradigm is the following: a word is represented by itself (*à, de* in Figure 7). A set of constraint can be specified between square bracket: for instance *[v]* stands for any verb, *[.pre]* for any preposition, *[vP]* for a verb at the present tense and *[<dire>v]* for a verb of infinitive form dire. It thus leads to the following FSA:

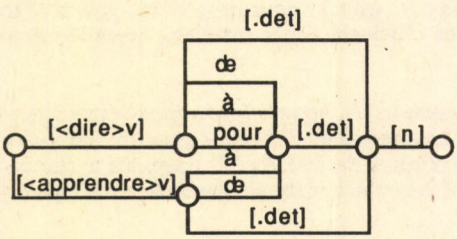


Figure 7

5. The kind of FSA of figure 7 is not easy to manipulate directly in formal operations.

To confront this FSA more easily to a morphologically tagged text, each transition is developed into a sequence of atomic symbols, each standing for a single peace of information. For instance [<dire>v] is translated into the FSA of figure 8. Each of these automata follows a grammar which can be specified by an automaton, actually the one of figure 9.

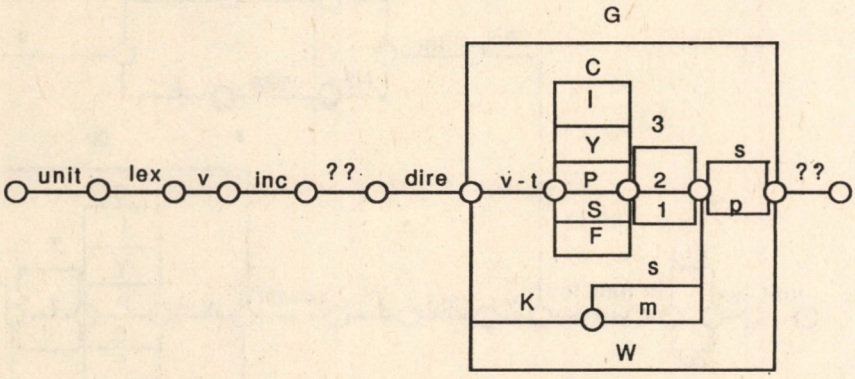


Figure 8[5]: translation of [<dire>v]: subpart of pattern A3

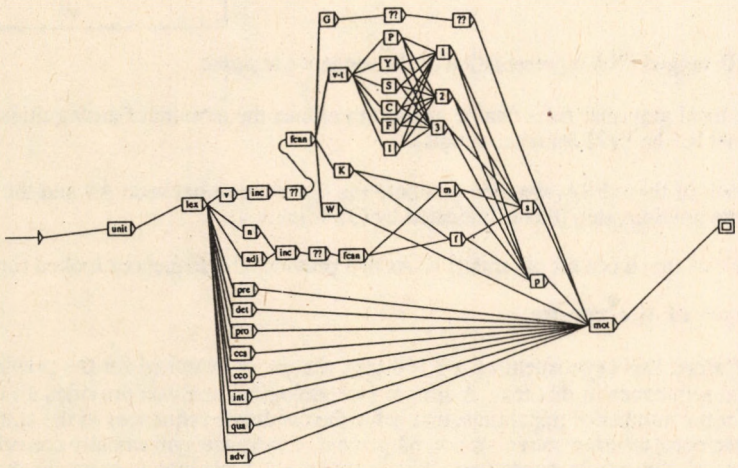


Figure 9: syntax of the representation of morphological information for french.

At this point we have the FSA A3, a text and a morphological dictionary. For each sentence of the text, the following computations are performed dynamically.

1. For each word, we look up the morphological dictionary (actually two dictionaries: one of simple words (600,000 entries) and one of compound words (150,000 entries)).

2. We build a FSA that represents the morphological ambiguïties and homographies.

---

[5]The abbreviations respectively have the following meanings: m:masculine, f:feminine, P: present, S: present of subjunctive, Y:imperative, F:future, C conditional, S passe simple s: singular, 1: first person, K:past participle, W:infinitive, G:present participle ..

This FSA agrees to the formal representation described by the FSA of figure 9. Thus it has the same format as the pattern A3. We give an example in Figure 10. Consider the sequence of two words *le passe*, which is ambiguous: *pass it* or *the key pass*, it is represented by :

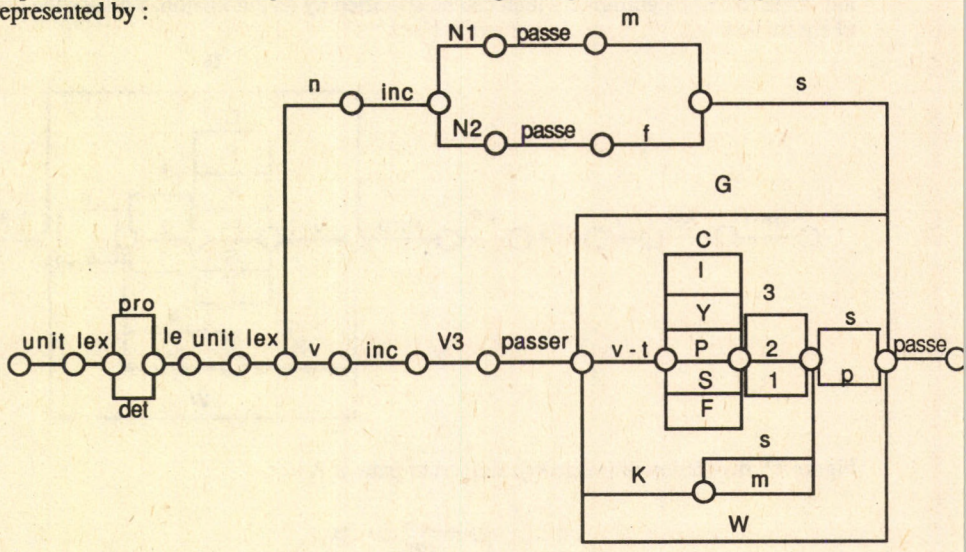


Figure 10: tagged FSA representation of the sequence *le passe*.

3. Some local grammar rules can be applied to reduce the amount of ambiguïties (See Emmanuel Roche 1992 for more details).

4. For each of these FSA, we then compute the intersection between A3 and the FSA built at the previous step (this is indicated by (3)in figure 4).

The result of step 6 is a set of (small) FSAs that provides the sequences looked for.

7. Sample of the results

We performed this experiment on a 3 Mb text, that is we searched for the previously described sequences in this text. A simple morphological analysis provides an upper bound for the number of matchings, that is for the candidate sequences to the structure V-Complement, whether correct or not. 62 percent of them are syntactically correct. We then applied the whole procedure, that is when we took into account each verb argument structure, we obtained a more precise result: 86% of the sequences were matched and 83% of the matched sequences were correctly recognized (the irrelevant ones being sequences like those of examples (1) and (2)).

The time spent on a IBM PS2 386 25Mhz under OS2 system was 1h25 for the morphological analysis and 2h05 for the analysis that took into account the argument structure. These figures include the dictionary lookup time.

8. Bibliography

Aho, Alfred V. , John E. Hopcroft, Jeffrey D. Ullman, 1974. The Design and Analysis of Computer Algorithms. Addison Wesley, 467p.

Boons, Jean-Paul ; Alain Guillet ; Christian Leclère 1976a. *La structure des phrases simples en français. I Construction intransitives*, Genève : Droz, 377p.

Boons, Jean-Paul ; Alain Guillet ; Christian Leclère. 1976b. *La structure des phrases simples en français. II Construction transitives*: Rapport de recherche du LADL, N°6, 85p.,tables et index, 58p. Paris :Université Paris 7.

Church, Kenneth, William Gale, Patrick Hanks, Donald Hindle, 1989. Parsing, Word Associations and Typical Predicate-Argument Relations. Internal report. Bell Laboratories, Murray Hill.

Courtois, Blandine, 1984, 1989. DELAS: Dictionnaire Electronique du LADL pour les mots simples du français, Paris: Rapport technique du LADL, Université Paris 7.

Gross, Maurice. 1968 Grammaire transformationnelle du français, 1-Syntaxe du verbe. Cantilène, Paris, 183p.

Gross, Maurice, 1975. *Méthodes en syntaxe*, régime des constructions complétives, Paris : Hermann, 415p.

Koskenniemi, Kimmo, 1990. Finite-State Parsing and Disambiguation. Coling-90. Proceedings of the conference. Helsinki.

Peireira, Fernando C.N. , Rebecca N. Wright. 1991. Finite state approximation of phrase structure grammars, 29th Meeting of the A.C.L, Proceedings of the conference. University of California, Berkeley.

Revuz Dominique, 1991. Dictionnaires et lexiques, méthodes et algorithmes. PhD dissertation. Université Paris 7, Paris, 130p.

Roche Emmanuel, 1992. Text disambiguatinon by finite state automata, an algorithm and experiments on corpora. COLING-92. Proceedings of the Conference, Nantes.

Rimon, Mori, Jacky Herz, 1991. The recognition capacity of local syntactic constraints. Fifth Conference of European Chapter of Association for Computational Linguistics. Proceedings of the Conference, Berlin.

Silberztein Max, 1989. Dictionnaires électroniques et reconnaissance lexicale automatique. PhD dissertation, Université Paris VII, Paris, 175p.

Acquiring and Representing semantic information from place taxonomies

ADRIANA ROVENTINI

This paper describes a preliminary investigation into the acquisition of semantic information from two Italian machine readable dictionaries with the aim of representing it in a Lexical Knowledge Base being implemented within the framework of the ESPRIT project ACQUILEX. The analysis focuses on the general and distinctive features of sets of words denoting places. Semantic and morphological characteristics of these sets are analyzed and a possible reorganization into meaning types and lexical rules, according to the lines of research of the project, is considered.

Introduction

Place is a very general concept, and, in any dictionary, various meanings will be associated with it and many words belonging to different lexical fields will be put in relation with one or more of them. In the dictionaries we are considering: the Italian Machine Dictionary and the Garzanti Italian Dictionary (henceforth DMI and GRZ), the first sense of 'luogo', which is the best Italian equivalent for the English place, is:

1) parte di spazio idealmente o materialmente delimitata
   part of space ideally or materially delimitated

from which it is easy to deduce that, because any experience or object we can think or speak about exists in some ideal or materially delimited space, the concept of place will spread out over the dictionary in many different directions and will belong, as an attribute, to many different taxonomies. The other meanings given in our dictionaries are the following:

2) parte della superficie terrestre	part of the earth's surface
3) edificio o parte di esso	building or part of it
4) parte di un oggetto, punto	part of an object, point
5) passo di uno scritto	written passage
6) momento opportuno	right moment

To which the DMI adds:

7) luogo geometrico	locus
8) condizione sociale	status

This set of meanings evidences the two essential dichotomies regarding the concept of place: the first opposing ideal or abstract places to material or concrete ones, the second opposing place as the typical object of the physical and geographical investigation and place as the theatre of human activity as well as the repository of

all human artifacts. In order to limit this investigation within such a broad and complex lexical domain, in this paper we will consider a set of places in which we find an intersection between the lexical types of both place and food, while the whole set of word senses defined as place is outlined only in its more general features. This selection is due to the fact that food is the lexical subset experimented for representation in the LKB within the ACQUILEX project.

In the dictionaries we analyzed 'luogo' as genus term, selects 668 word senses from the DMI and 278 word senses from the GRZ (where the difference in number can be attributed to the difference in size) and, in order to retrieve the most general associated features, this first level of hyponyms was considered sufficient. Using the set of definitions as a corpus, semantic information, usually placed in the differentia part or 'differentia specifica', was extracted from them. This kind of analysis was essential in order to capture the different semantic features concerning the principal meaning types of place which could be represented in a more explicit and structured way either into templates following the example given by Calzolari (1991) for the concept of substance, or, when possible, by means of lexical rules as Briscoe and Copestake propose in (1991a, 1991b). They present a definition of lexical rule formalized within the framework of unification based approaches to the lexicon which employ typed features structures and default inheritance. They also argue that, in the lexicon, sense extension and morphological processes of derivation and conversion are similar enough to be treated as lexical rules and that the distinction between these phenomena is not as sharp as usually meant. On this purpose, the following phenomena were considered, when analyzing the metalanguage of the definitions:
a) the set of nouns connected with the genus;
b) the types of functions which link the genus to the differentia;
c) the properties that, in the differentia, are associated with the genus;
d) the connections between morphological and semantic features.
Within the corpus of the definitions, almost any of these phenomena are related to typical formulae or syntactic structures.

Types of place and sense extension

A typical procedure adopted by lexicographic metalanguage is to join the genus with another noun which is either an instantiation or an extension of it. This phenomenon is due to the necessity of compressing the information in printed dictionaries, from which it descends that, within a same definition, the instantiation of a meaning type is usually evidenced by a disjunction or by simple juxtaposition, while the sense extension is rather revealed by conjunction, as is exemplified below:

(a) bassofondo: luogo o quartiere della malavita
slum: place or neighbourhood of criminal underworld
(b) galera: luogo e situazione in cui si soffre
gaol: place or situation in which one suffers

In example (a) is given a first and more general indication, 'luogo' followed by an instantiation or specification of it 'quartiere' (neighbourhood). In the second example (b), what is given is not a type of place but a metonymic extension of it: the condition of being in a certain place, in this case the condition of being in prison. Unfortunately, owing to the lack of consistency of the metalanguage, this use of conjunction or disjunction is unreliable; it was anyhow useful as cue of a typical phenomenon very frequent in this subset and worthy of a punctual analysis which

pointed out the most common meaning types as well as the cases of sense extension by metonymy. The following are the cases of this kind of sense extension found in the whole subset of definitions:

luogo e azione (place and act of) as in: attracco, imbarco (docking, embarcation)
luogo e modo (place and way) as in: collocazione (collocation)
luogo e collezione (place and set of things) as in: utensileria (tool room)
luogo e persone (place and set of people) as in: curia, cantoria (curia, choir)
luogo e arte di (place and art of) as in: carpenteria, pasticceria (carpentry, confectionery)
luogo, carica e durata (place, office and tenure) as in: assessorato (councillorship)
luogo coltivato e le piante (place and set of trees) as in: oliveto (olive grove)
luogo e quantita' (place and quantity) as in: fungaia (mushroom bed)
luogo, mobile e insieme (place, furniture and set of) as in: guardaroba (wardrobe, cloakroom)
luogo e tempo (place and time) as in: esilio (exile)
luogo e situazione (place and situation) as in: galera (prison)
luogo e animali (place and set of animals) as in: stalla, porcile (cowshed, pigsty)

It is possible to add some more combinations but, as far as semantic information is concerned, these are sufficient to evidence a lexical phenomenon which is relevant from the point of view of its encoding within a lexical component for Natural Language Processing systems because of the change of the subcategorization features involved when shifting from concrete to abstract (e.g. place/art), from inanimate to animate (e.g. place/plants, place/animals), or from inanimate to human (e.g. place/people).
With regard to sense extension, it is interesting to note that certain types of places, for example those in which animals live or are kept and raised, in addition to the metonymic, show the metaphoric sense extension. In fact the metaphoric extension animal/human, evidenced by Briscoe & Copestake (1991b) is also productive within this lexical type. Words like 'pollaio' (poultry pen), 'stalla' (cowshed), 'porcile' (pigsty), 'nido'(nest), 'formicaio' (anthill), 'alveare' (beehive), 'tana' (den) and so on, can be used to denote places inhabited or crowded by human beings and, with the same of association, we can say 'This room is a pigsty' just as we say 'Luigi is a pig'. In the subset considered this phenomenon is regular enough to be dealt with by a lexical rule. of the kind animal metaphor described in the above cited paper.

Functions, properties and constituents

When we went to analyze the differentia in our set of definitions these types of semantic information were considered relevant:

a) the aim or goal to which a place is devoted;
b) the properties which outline it in its appearence or formal side;
c) the constituents: items, structures, plants, animals which characterize it.

All this semantic information, in the general template for place (see the figure 1), is gathered under three main attribute tags: ***function, property, constituency,*** which, in a more general and comprehensive way, respectively correspond to the telic, formal and constitutive roles described by Pustejovsky.
The pure locative function is obviously inherited from the genus and is a basic value within the entire set of definitions. This is joined, in most of them, with telic functions

evidencing the particular aim or goal to which a place is devoted. The metalanguage expresses the telos in different ways ranging from a simple adverb or preposition to more complex prepositional phrases, as appears in the examples below:

1a)	luogo dove si allevano	place where....are raised
1b)	luogo in cui si allevano	place in which...are raised
1c)	luogo ove si allevano	place where....are raised
1d)	luogo di allevamento	place of rearing
1e)	luogo per l'allevamento	place for rearing
2a)	luogo adibito alla raccolta	place assigned for gathering
2b)	luogo di raccolta	place of gathering
2c)	luogo ove si raccolgono	place where... are gathered

In these cases one telic tag will be able to substitute different but equivalent formulae. The same is worth for the purposes which have a same semantic value (the rearing or the gathering) but are defined using either a verb phrase as in 1a,1b, 1c, 2c, or a noun phrase formed by the corrisponding deverbal noun as in 1d, 1e, 2a and 2b.

A second type of function frequently combined with place is that of motion. Several places have the character of being reference marks with this regard. There are places from which, or towards which, or across which the motion is directed, and all these are characterized by the joint locative-motion function. Most of these typical linking formulae are listed in the templates under ***function*** attribute.

As regards the properties associated with place most of them come from definitions which are very simple in their structure and usually consist of the genus plus one or more modifiers: adjectives or past participles with an adjectival value. The following are a few examples of this kind of definitions where the genus is described in its purely formal appearence:

luogo caldo	warm place
luogo squallido e sporco	dreary and dirty place
luogo abitato	inhabited place
luogo montuoso	mountainous place

By means of a careful analysis of the whole set it was possible to acquire a number of this kind of semantic features and to gather them using a few tags under the generic attribute ***property***. Obviously not all the modifiers fall into one of these features, but certainly most of them do.

An example of a typical formula revealing the constitutive argument of a noun, ***constituency*** in the Template, is shown in the definition below:

sterpaia: luogo pieno di sterpi
scrub : place full of thorns

Lexical regularities and morphological features

The hyponyms of place show some interesting regularities either in their word formation or in their lexical features. The suffixes which form homogeneous sets of words denoting places are the following:

-ETO	-ETA
-AIO	-AIA
-OIO	-ILE
-ERIA	-ORIA
-TURA	-ARIO

and also

-FICIO	-COLTURA

as second element of many compounds.
These sets mostly refer to particular types of places which exhibit these defining patterns:

il luogo coltivato e le piante	the cultivated place and the plants
luogo coltivato a/ad	cultivated place for
luogo dove si allevano	place where.... are/is raised
luogo dove si conserva/no	place where....are/is kept
luogo dove si fa o si vende	place where....are/is made or sold
luogo folto di	place thick with..
luogo in cui crescono	place where.... grow
luogo ricco di	place rich in
luogo piantato a	place with a plantation of..
luogo pieno di	place full of

Because these formulae define words denoting places where:
edible plants are cultivated (such as terreno,campo, bosco, coltivazione, piantagione);
animals are grown (such as vivaio, allevamento);
animals or vegetables are transformed into food, (such as fabbrica, laboratorio);
food is eaten and/or sold (such as negozio, bottega, locale);
the search was deepen in this direction in order to select homogeneous subsets of words which could be put in relation with lexical types of food in the LKB. Among these hyponyms we choose to set the template of 'terreno' (see figure 2) in order to show a set of features representing a subtaxonomy of place.

Lands, fields, woods and cultivated areas

These types of places are characterized by the following suffixes: -ETO/A, -AIO/A. A few examples are: 'castagneto, acereta, oliveto, erbaio, favaio, zuccaia, poponaia'. The suffix -ETO/A is the most interesting because it carries an univocal meaning and could be dealt with by a lexical rule which specifies that every noun of tree, plus the suffix -ETO, (in a few cases -ETA), forms another noun denoting either the place or, by extension, the set of plants growing there.
The other suffix works in the same way in this subset, but also forms words denoting place + animals such as: 'colombaia, pollaio', and place + quantity or set of materials such as 'legnaia, ghiacciaia, granaio'. Furthermore it enters into the formation of nomina agentis such as 'macellaio, verduraio, lattaio'.
Two other sets of words homogeneous in meaning and word formation are constituted by hyponyms of 'piantagione, coltivazione and allevamento'. Both use the noun COLTURA as the second element of compounds denoting cultivations or rearings from a technical point of view and all these hyponyms are regularly defined as follows:

a) apicoltura: allevamento delle api (the keeping of bees)
b) viticoltura: coltivazione della vite (the growing of grapes)

The first word sense of 'allevamento e coltivazione' in Italian is 'art of..' but they also denote the place in which the cultivation or the rearing are performed and the set of plants or animals grown. It derives that, in some contexts, the meaning of words like 'apicultura' can shifts to denote the place where the 'art' is carried on.

Factories and shops

The places in which various artifacts are made or sold are derived from nouns by means of the suffix -ERIA or, especially in the case of factories, we can find compounds in which the second element -FICIO is the carrier of the univocal meaning: 'factory of..'. This second type of word formation is interesting because it is very productive and, as -ETO, could be manageable by means of rules. On the other hand -ERIA, even if productive in this lexical field, as it produces many other different formations with regard to meaning, would be difficult to manage by lexical rules. In this subset it forms nouns denoting shops, and in some cases, by extension, the set of goods sold which there are sold and, in some cases also the art of making such artifacts. An example of that is the noun 'pasticceria' (confectionery).

Final remarks

It is interesting to highlight that most of the suffixes which form nouns denoting places (and extensions of them) do not give rise to changes in grammatical categories. They derive denominal nouns in which only the meaning is involved in the change. For this reason, they could be seen as a kind of marked sense extension and, in some cases, be handled by lexical rules (see the cases of -ETO and -FICIO); nevertheless, given that most of them do not carry univocal meanings but, on the contrary, derive various types of word senses and are productive in many different lexical domains, in our opinion it is not convenient to manage them by lexical rules. By means of lexical rules,instead, should be treated a phenomenon as sense extension by metonyny which, as shown in this paper, is very frequent throughout the various levels of the place taxonomy. Frequent, in the subsets analyzed, is the case in which an inanimate noun denoting a place, is used in an animate, collective sense denoting a group found within it. In these cases it is worth finding a lexical rule for the general sense extension inanimate noun - animate collective noun and domain-specific sub-rules could then be set up to capture the various types within this category.

References

Boguraev B.K., Briscoe E., Calzolari N., Cater A., Meijs W., Zampolli A. (1988), 'Acquisition of Lexical Knowledge for Natural Language Processing System', proposal for ESPRIT BRA, Cambridge (UK).

Briscoe E. and Copestake A. (1991b), 'Sense extension as lexical rules', Proceedings of the IJCAI Workshop on Computational Approaches to Non-Literal Language, Sidney, Australia. ESPRIT BRA-3030 ACQUILEX Working Paper No.022.

Calzolari N. (1991), 'Representation of semantic information in a Lexical Knowledge Base', in Proceedings of the ACL SIGLEX Workshop on Lexical Semantics and Knowledge Representation, Berkeley, California, pp. 188-197.

Copestake A. and Briscoe E.(1991a), 'Lexical Operations in a Unification-based Framework', Proceedings of the ACL SIGLEX Workshop on Lexical Semantics and Knowledge Representation, Berkeley, California, pp.88-101. ESPRIT BRA-3030 ACQUILEX Working Paper No. 021.

Ostling A. (1991), 'Sense Extensions in the Italian Food Subset' ACQUILEX, ESPRIT BRA-3030, Working Paper No. 024, ILC-December-1991, Pisa. pp.88-101. ESPRIT BRA-3030 ACQUILEX Working Paper No. 021.

Pustejovsky J.(1991), 'The generative lexicon' in Computational Linguistics, 17(4).

Fig. 1 TEMPLATE for PLACE (and related defining patterns)

	Function
Addetto a	Intended for
Adibito a	Used as
Assegnato a	Assigned to
Attrezzato a	Equipped with
Destinato a	Intended for
Designato per	Appointed for
Coltivato a	Cultivated with
Piantato a	Planted with
Che serve a	Necessary for
Che + VP	Which + VP
Di+ NP	Of + NP
Dove si + VP	Where + VP
In cui si + VP	In which + VP
Ove si + VP	Where + VP
Per + NP	For + NP
Da cui + VP	From which + VP
Donde si + VP	" Where + VP
Attraverso il quale si + VP	Through which + VP
Per cui si + VP	
Per dove si + VP	
Dove si + VP	Where + VP
	Property
Larghezza	Width
Altezza	Height
Profondita'	Depth
Forma	Shape
Struttura	Structure
Popolamento	Population
Posizione	Position
Esposizione	Exposure
Clima	Climate
Temperatura	Temperature
Umidita'	Humidity
Coltivazione	Cultivation
Costruzione	Building
Apparenza	Appearance
	Constituency
Ricco di	Rich in
Pieno di	Full of
Folto di	Thick full of
Coperto di	Covered with

Fig. 2 TEMPLATE for TERRENO (And related defining patterns)

	Function
Adibito a	Used as
Destinato a	Intended for
Pronto per	Ready for
Adatto a	Suitable for
Coltivato a	Cultivated with
Lavorato a	" "
Piantato a	Planted with
Tenuto a	
	Property
Esposizione	Exposure
Struttura	Structure
Altezza	Height
Forma	Shape
Ampiezza	Width
Posizione	Position
Coltivazione	Cultivation
Umidita'	Humidity
	Constituency
Coperto di	Covered with
Folto di	Dense with
Pieno di	Full of
Ricco di	Rich in
Ricoperto di	Covered with

On using the french lexicon-grammar in a French-English bilingual dictionary

MORIS SALKOFF

The problem of finding the English equivalent of a French verb clearly involves the context around the verb, but it does not seem possible to order the equivalents by that context. I have used the French lexicon-grammar, with its natural systematic ordering of the verb uses in French, as the basis for ordering the translational equivalents in English. Hence I have listed these equivalents in the order of the French verb uses; the systematicity is in the ordered list and the irregularity is confined to the choice of equivalent.

I then pose some interesting questions about this procedure and formulate an experiment to test the answers proposed. It turns out that three semantic noun sub-classes: Na, Nh and Nconc, suffice for finding the equivalents of most of the verb uses of the verbs starting with the letter A. The question then remains whether the additional semantic sub-classes that I have introduced, which number about 25, will be enough the specify the equivalents for the remaining verbs of the lexicon-grammar.

On using the lexicon-grammar in a bilingual dictionary

One of the most vexing problems encountered when attempting to write a program of automatic translation is that of resolving the polysemy of most words. A simple inspection of the chaotic presentation of the translations recorded in any bilingual dictionary will convince one that it would seem impossible to impose an order on the arbitrary translations of any given entry. Some translations are imposed by a particular syntactic context, e.g., *abuser*:

(1)a Max abuse --> Max overdoes it
b Max abuse de N --> Max takes advantage of N

Here, each context contains a particular object of *abuser*: when *abuser* is followed by 0, the translation is *overdo*; when it is followed by *de N*, it is *take advantage of*.

However, many other translations are due to a semantic context containing particular subclasses of words, e.g., *absorber*:

(2)a Na (Le travail) absorbe Nh (Max) --> Work engrosses Max
b Nc (Le buvard) absorbe Nc (le liquide) --> The blotter absorbs the liquid
c Nh (Max) absorbe Nc (le boisson) --> Max imbibes the drink

The problem is made even more difficult if it is expressed in terms of semantics, i.e., by saying that most words have many meanings which are hard to distinguish by formal means. Nevertheless, examples (1) and (2) point up the essential problem: how is one to choose the meaning of a polysemous word, based on the information furnished by a polysemous context?

Lehrberger & Bourbeau (1988, pp. 114-116) for example, point to the many senses of the English preposition *on* as a difficulty that it may not be possible to overcome without further advances in semantic analysis. Their example is instructive, and will repay looking at. Let us examine the French equivalent of this problem: the prepositional phrase *sur N*.

In general, the translation of any prepositional phrase is closely linked to the context in which it is found: the translation varies according as it is in a sentence adjunct, for example, or in the object of a verb. Consider the translation of the prepositional phrase *sur ce point*:

(3)a Paul embête Max *sur* ce point --> Paul bothers Max *about* this point
b *Sur* ce point, Paul est inflexible --> *On* this point, Paul is inflexible

In (3)a, *sur ce point* is in the object of *ennuie*, so that *sur* translates as *about*. In (3)b, *sur ce point* is a sentence adjunct, and a better translation of *sur* is *on*.

Let us now examine the translation of the phrase *sur N* in the object position as the noun *N* and the verb vary.

(4)a (La maison) donne *sur* (le jardin) --> (The house) looks out *on* (the garden)
b (Le pion) avance *sur* (l'échiquier) --> (The pawn) advances *on* (the chessboard
c (L'armée) avance *sur* (l'ennemi) --> (The army) advances *against* (the enemy)
d (Paul) l'emporte *sur* (Max) --> (Paul) prevails *over* (Max)
e (Paul) interroge Max *sur* (ce point) --> (Paul) interrogates Max *about* (this point)
f (Paul) pose le verre *sur* (la table) --> (Paul) sets the glass *on* (the table)

There are two ways of looking at this array of varying translations for *sur*. In the first instance, one can declare *sur* to be highly polysemous, since it can be translated as *on*, *over*, *against*, etc. Furthermore, each such translation is arbitrary, so that no formal rules could be constructed that would decide upon the proper translation. The conclusion then follows that automatic translation of polysemous words is an intractable problem that cannot be handled in any systematic fashion.

In the second instance, however, we note that in each of the above sentences, the decision as to the required translation of *sur* can be taken on the basis of the syntactic and semantic context of *sur*. In sentence a, *sur* is in the prepositional object *sur N2* of *donner*, and its translation is *on*. In sentences b and c, the same object *sur N2* appears after *avancer*, but here the translation of *sur* depends on the subclass of *N2*: in b, *N2* is a concrete noun,; and in c it is *Nh*, a 'human' noun. The same prepositional object in sentence d after *emporter* requires that *sur* be translated as *over*. In sentence e, the object of *interroger* is *N1 sur N2*, and the translation of *sur* is *about*. However, the same formal object in sentence f for *poser* leads to a translation *on* for *sur*.

We see that the problem here of the apparent multiple meanings of *sur* can be replaced by an equivalent one that is easier to handle, viz., finding formal definitions of each of the contexts in which *sur* can appear. This in turn suggests that it may be possible to utilize the dependence of translation on context for handling the question of polysemy in a formal, systematic way. At first sight, this dependence on context seems to be linked both to the multiple meanings of individual words and to the multiple meanings of the surrounding context induced by the polysemy of the words appearing in that context. The resolution of the problem of polysemy thus seems *a priori* extraordinarily complex, involving the simultaneous juggling of polysemous words and polysemous contexts.

However, a detailed compilation of verb uses in French that has recently been completed at the L.A.D.L. may be of help in this question. This compilation shows us that relatively few syntactic and semantic sub-classes of nouns, verbs, etc. are required to describe verb uses in French, at least in a quite general way. With the help of this compilation, it is now possible to search *systematically* for the translation of each verb use in the context 'verb + object' in order to choose the best one. In so doing, we shall also see how many semantic sub-classes of nouns are required to carry out this project.

The study of verb uses proposed above amounts to a direct and systematic examination of the problem of polysemy. For each verb use is defined by a sentence containing the verb, its subject and its complements, so that the detailed listing of verb uses in effect makes available all the sentence contexts in which a given verb might be found. Suppose that one were to try to translate all these verb uses into English, for example. How many different translations would be required for each verb, and what would allow us to determine the choice of translation? The number of different translations that would be required is a measure of the polysemy, and the resolution of the polysemy lies precisely in choosing an appropriate translation for each verb use.

An examination of the polysemy in this fashion constitutes an experiment in French-English translation. The variables are the semantic sub-classes of nouns required to describe the verb uses of French, and the translation equivalents of each verb use. Each verb use is an entire sentence, e.g.,

(5) Max abaisse Luc à demander 100fr --> Max humbles Luc into asking for 100fr

The syntactic analyzer produces an analysis of such a sentence using the information that both the subject and object of *abaisser* (*humble*) must be 'human' nouns, for this verb use. The object can be a 'concrete' noun in another use of *abaisser*, but then the translation is different:

(6) Max abaisse le rideau --> Max lowers the curtain

In this way, both the syntactic analyzer and the translation program make use of the sub-classes of nouns introduced to describe the verb uses. The experiment can now be set up in the following way.

(i) I first translate a portion of the lexicon of verb uses of French, say all the verbs beginning with *a*, taking care to use the minimum number of different English equivalents needed to translate correctly all the verb uses of a given verb. In certain cases, we shall see that this means using an approximate translation in order not to increase the number of different translations.

(ii) Many provisional semantic sub-classes of nouns are required in order to carry out the translations of the previous step. Some such classes are already in the string grammar of French, and are used by the restrictions of the grammar to eliminate incoherent analyses. Three of the the most important of these classes are the following:

(7) Nh human nouns:Max, man, person; jury, council.
Na abstract nouns: despair, construction
Nc concrete nouns: table, lamp, chair

These are quite approximate semantic classes, and a classification of nouns using only these three sub-classes amounts to a rough and ready partition of nouns. Nevertheless, for many verbs, such a tripartition is sufficient to separate the translations of verb uses. Thus, I find the following translations to be adequate for the verb *abattre*:

(8)a Nh (Max) abat Nc (maison) --> Max demolishes the house
b Nh (Max) abat Nh (Luc) --> Max kills Luc
c Nh (Max) abat Na (travail) --> Max zips through work

It turns out that a classification of nouns based on these three classes: Nh (human), Nc (concrete),and Na (abstract) suffices to separate the translation equivalents for most of the verb uses. Note that this result is not a priori predictable, e.g. from theoretical considerations alone.

(iii) For some verbs, particular semantic sub-classes of nouns are required in order to separate the translation equivalents. Certain additional noun sub-classes have already been introduced into the verbal classification carried out at the LADL. For example, the following two sub-classes discriminate certain verb uses:

(9)a Nt text nouns: book, report, manuscript,...:
b Np psychological nouns: mind, spirit, wits,...

They appear in the following types of verb complements:
(10)a Max accroche au rapport que Ph --> Max appends to the report that S
b Cet effort ancre dans son esprit que Ph --> This effort fixes it in his mind that S

These are the contexts for which Np and Nt were originally defined; they are useful however in other verb uses, where they separate the translation equivalents:

(11)a Max annonce le gagnant --> Max announces the winner
b Ce résultat annonce une intelligence (Np) --> This result gives sign of intelligence (not: *announces an intelligence)
c Max accolade Marie --> Max embraces Marie

d Max accolade le mot (Nt) --> Max brackets the word (not: *embraces the word)

Here, the difference between *Nhum* and *Np* or *Nt* allows us to choose the appropriate translation of *announcer* and *accolader*.

Other semantic classes I have found useful include the following:

(12)a Nm metal: armer Nc(poutre) de Nm(fer) --> brace the beam with iron
b Nn sound: Nn(bruit) abasourdit Nh --> Noise deafens Max
c Ns measure: Nh(Max) accuse 85 kg. --> Max professes to be 85kg.
d Nv vehicle: Nh accroche Nv(voiture) --> Max runs into the car
e Nw weather: Nw(tempête) s'annonce --> A storm is brewing

I have carried out these three steps for the 400 French verbs beginning with *a* whose uses have been studied in the LADL, and have found that about 25 semantic sub-classes similar to those in (12) are required to separate all those translations which cannot be distinguished by the use of only the three classes Nh, Nc and Na. The semantic subclasses are presented in Annex 1, and excerpts of the translations of verb uses are given in Annex 2.

The semantic translation classes are defined, approximately, in the following way. Consider a verb use that contains one of these semantic classes in subject or object position, e.g., Na (abstract noun). Thus the verb *abasourdir* is translated as *dumbfound* when the subject is an abstract noun Na. This translation is different from the translation of another use of *abasourdir* where the semantic sub-class 'sound', Nn, appears in the subject, as in (12)b. In that case, *abasourdir* translates as *deafen*. This difference in translation defines the sub-class 'sound', Nn. A word is classified in that semantic sub-class (Nn) if the translation of the verbs appearing with it is the one associated with that sub-class.

Note that these semantic sub-classes are not defined absolutely, by reference to their meaning in isolation. Rather, they are defined by a difference in meaning between two translations. With one such semantic sub-class occupying the verb or object position of a given verb, a verb use is defined. Given two such verb uses, we frequently obtain two different translations. Then the semantic sub-class is defined by the translation required for the verb use containing that sub-class.

The problem of how to handle polysemy is now replaced by another, perhaps more tractable question: to what extent is it possible to obtain a coherent classification of nouns using the semantic sub-classes defined above in terms of translation equivalents? Posing the question in this way amounts to asking whether the differences in meaning, under translation, of a given verb can be captured by a classification of the nouns appearing with it. On the other hand, trying to characterize the polysemy of a verb directly amounts to attempting to define its absolute meanings, which is considerably more difficult.

Results

It turns out that the syntactical differences among the verb complements of a given verb use, taken together with the rough classification of nouns according to the three classes Nh, Na and Nc is often sufficient to separate all the translations of the uses of a verb. Thus, the verb *s'acharner* can be separated in this way:

(13)a (Nh, Na) s'acharne contre, sur Nh --> Max, fate hounds Luc
b Nh s'acharne à, sur Na --> Max slaves at his work

c Nh s'acharne à faire cela --> Max is bent on doing that

Note that the difference between (13)a and b, when the preposition is *sur*, is the difference between Nh and Na; this is enough to differentiate the translations. The translation for (13)c is obtained by noting the difference in the complement: the complement is *Prep N* in (13)a and b, but *à V O* in (13)c.

For some verbs, the translation does not change with the change in complement:

(14)a Nh acclame (Nh, Na) --> Max acclaims (Luc, the suggestion)
b Nh acclame Nh No --> Max acclaims Luc emperor
c Nh acclame Nh de faire cela --> Max acclaims Luc for doing that

Here, No is a noun of 'function': president, king, emperor, ambassador, etc. However, its exact nature is of no concern here, since the translation is invariable.

Similarly for *acculer*:

(15)a Na accule Nh à Na --> This event drives Max to despair
b Na accule Nh à faire cela --> This event drives Max to do that
c (Na, Nh) accule (Nh,Nc) contre Nc --> (Luc, the force) drives (Max,the box) against the wall

Various semantic noun-sub-classes appear in the verb uses of other verbs, but it is frequently possible to avoid having to depend on them for a decision as to the required translation. Sometimes, it is the special nature of the verb complement that allows us to make the right decision:

(16)a Nh avale N --> Max swallows N
b Nh avale que Ph --> Max accepts that S
c Nh avalise N --> Max endorses N
d Nh avalise que Ph --> Max backs it that S

For both of these verbs, the difference in translation depends on the nature of the verb complement. In each case, the noun phrase complement accepts only a limited sub-class of nouns. The verb *avaler* is usually followed by a noun indicating food or drink, and the verb *avaliser* by a noun like *check*, or certain abstract nouns. However, the difference between the verb complements is enough for choosing the correct translation, so that a more precise specification of the noun phrase is unnecessary.

The verb *atteindre*, on the other hand, has a different translation for various noun sub-classes when its object is a noun phrase.

(17)a Nh atteint Nl (ville) --> Max reaches the town
b Nh atteint Nc (boite) --> Max reaches for the box
c (Nh,Na) atteint Nh --> (Luc,disaster) overtakes enemy
d Nh atteint Nua (cinquantaine) --> Max is getting on to 50
e Nh a atteint Nu (65) --> Max reached 65
f Naa (cancer) atteint Nj (poumons) --> Cancer affects the lungs
g Naa (oreillons) a atteint Nh --> Max caught the mumps

The following noun sub-classes are needed, in addition to Nh, Na, and Nc:

(18) Naa : disease (cancer, mumps)
Nl : place (town, room, street)
Nu : time (year, week)
Nua : approximate time (20s, 30s)
Nj : body part (lungs, arm, leg)

The most difficult class to define is Nl, nouns of 'place'; in (17)b, the box can either be a concrete object, or a place where something is to be put. The translation of *atteindre* depends on this interpretation.

Special cases

(i) The passive participle

The French passive participle, whether in the noun phrase or in a passive sentence, is generally translated by the addition of the passive morpheme (*-ed, -en*, etc.) to the infinitive. This is the case for the majority of the verbs translated in Annex 2; no translation is shown for the passive participle for these verbs.

However, there are two cases where the passive participle is not translated by the addition of the passive morpheme to the infinitive. The first case occurs when there is no passive of the English translation, and the second when the French passive participle is adjectival in nature. The first type can be illustrated with *agenouiller* and *additionner*:

(19)a La peur agenouille Max devant le roi --> Fear makes Max kneel down before the king
b Max a été agenouillé devant le roi par la peur --> Max was brought to his knees before the king by fear
c Max additionne Nc de Nd --> Max makes a mixture of (mixes) Nc with Nd
d Nc a été additionné de Nd --> Nc was mixed with Nd

Since the English equivalent of *agenouiller* has no passive, the participle *agenouillé* must be specially handled. In the same way, if we use the translation *make a mixture of* for *additionner*, then the passive participle must be translated as *mixed*. A similar treatment is required for the following verbs:

(20)a *assiéger* --> *throng about*, *assiégé* --> *(was) beset*;
b *atteindre* --> *get, catch*; *atteint* --> *(be) stricken with*, etc.

The second case is the passive participle after *être (be)*, which is almost adjectival. Whether treated as an adjective or as a verbal form, it requires special treatment, e.g. *agripper*:

(21)a Max s'agrippe au balcon --> Max clutches at the balcony
b Max est agrippé au balcon --> Max is clutching the balcony

(ii) The reflexive *se*

The particle *se* does occasionally translate into the English reflexive *-self*:

(22)a Max s'arroge le droit de.. --> Max arrogates to himself the right to..
b Max s'est absenté aujourd'hui --> Max absented himself today.

However, it frequently requires special treatment, so that its appearance with any verb must be examined carefully. In the case of *abandonner*, for example, the syntactic context (the verb complements) determines the translation:

(23)a Max abandonne --> Max gives up
b Max s'abandonne --> Max lets himself go
c Max s'abandonne au désespoir --> Max gives way to despair
d Max s'abandonne à V O --> Max goes so far as to V O

For other verbs, the semantic subclass of the arguments (subject, object) may determine what translation is required, e.g., *abattre*:

(24)a (La température + le vent) s'abat --> (The temperature + the wind) subsides
b Le mur s'abat --> The wall falls down

(iii) Idioms

Studies by M. Gross (19xx) indicate that there are approximately as many idiomatic expressions as there are verb uses. Hence, the necessary inclusion of idioms in the dictionary will double its size. I have indicated the type of idiomatic expressions that have been collected by Gross for just the three verbs *abaisser*, *abandonner* and *abattre*. The entries for such idioms consist of particular words, not sub-classes:

(25)a Max abandonne la lutte --> Max gives up the struggle
b Max abaisse ses cartes --> Max lays down his hand
c Max abat atouts --> Max pulls trumps

The final results for the translations of the verb uses of verbs beginning with *a* are the following. For about 205 verbs, either the syntax of the object, or the use of the three subclasses Nh, Nc, and Na suffices to separate the translations. Another 118 verbs have only one possible translation, and so present no special problem. Thus, there remain about 80 verbs, out of the 400 beginning with *a*, that require the use of the semantic sub-classes in Annex I for translation, i.e., about 20% of the verbs.

Conclusions

Rather than speak of the polysemy, or multiple meaning, of words, it is more helpful to note that the translation of most words varies with the syntactic and semantic context surrounding them. The syntactic contexts can be ordered for French verbs by using the list of entries of the lexicon-grammar of French. Each entry corresponds to a separate verb use, so that this listing constitutes an ordering of the syntactic contexts of the verb. This ordering, in turn, represents the regularity of the French-English lexicon; the irregularity is confined to the translation of each item.

After this ordering of the verbs, there remain the semantic differences among the contexts. I have treated these by defining semantic sub-classes of nouns in an approximate fashion, in order to handle those syntactic contexts that differ only by some difference in the nouns appearing with the verb. The question now remains whether these sub-classes will suffice to handle the remaining verbs of the lexicon. Translating the remaining verbs of the lexicon-grammar will complete the partial experiment discussed here.

Annex I: Sub-classes

Na abstract	Naa sickness	Nab bad(disaster)	Nal law	Nam math term
Nb boat				
Nc concrete	Nca liquid	Ncb food	Ncd musical instr	
Nd color	Nda form			
Ne tool				
Nf machine				
Ng cards				
Nh human	Nha military	Nhb country	Nhc party;group	Nhd name
Ni animate				
Nj body part	Nja feature			
Nk Sing.Collective		Nka group		
Nl Place	Nla city;region	Nlb road		
Nm metal				
Nn sound				
No function				
Np psych.				
Nq particular nouns (for a given verb)				
Nr				
Ns measure				
Nt text	Nta word	Ntb sentence	Ntc Ling.element	Ntd music
Nu time	Nua Q-aine	Nub date		
Nv vehicle				
Nw weather				
Nx chem.prod.	Nxa acid	Nxb drug		
Ny money	Nya debt;tax			
Nz clothes				

Annex 2: Excerpts from the dictionary

Nh	abaisser	Nc de Q Ns	lower Nc by 5 (cm)
Nh	abaisser	Na (temp.;fever) de Q Ns	lower Na by 5 (degrees)
N	abaisser	Nh	humble Nh
N	abaisser	Nh à ce que Ph	humble Nh into Ving O
N	abaisser	Nh à V O	humble Nh into Ving O
Nc	s'abaisser	0	subside
Nh	s'abaisser	devant Nh	Nh humble -self before Nh
Nh	s'abaisser	jusqu'à V O	Nh stoop to Ving O
Nh	abandonner	Nh	desert
Na	abandonner	Nh	desert
Nh	abandonner	Nl (lieu)	leave (place)
Nh	abandonner	Nl à Nh	leave (place) to (Max)
Nh	abandonner	Na à Nh	leave (problem) to (Max)
Nh	abandonner	Na à Nh	leave (right) to (Max)
Nh	abandonner	0	give up
Nh	s'abandonner	0	let -self go
Nh	s'abandonner	à Na	give way to (despair)
Nh	s'abandonner	à V O	go so far as to V O
Na	abasourdir	Nh	dumbfound
Nn	abasourdir	Nh	deafen
N	abatardir	Nh	mongrelize Nh
N	abattre	Nh	kill
N	abattre	Ni	slaughter (cow)
N	abattre	Nc	demolish (house)
N	abattre	Ng	lay down (spade king)
Nh	abattre	Na	zip thru (work)
N	abattre	Nc de, sur Nc	cut off (pear) from (tree)
N	abattre	Ny de Na	cut off (Ifr) from (taxes)
Nh	s'abattre	0	become depressed
Na	s'abattre	0 (wind, heat,...)	subside
Nc	s'abattre	0 (bomb, pole)	crash down
Nh,Ni	s'abattre	sur Ni (Max,dog)	pounce on (cat)
Na	s'abattre	sur Nh (insults)	sweep down on
Nw	s'abattre	sur Nh,Nc (storm)	sweep down on
Nh	abdiquer	0+Nq	abdicate (0+crown, throne..)
Nh	abdiquer	Na devant Nh	renounce (right) before Nh
N	abimer	Nc	ruin (house)
N	abimer	Na	ruin (work)
Na	abimer	Nh	overwhelm (Max)
Nh	abimer	Nh de Na	overwhelm Nh with (insults)
Nb	s'abimer	0 (boat)	sink
Nc	s'abimer	0 (coat, house)	get ruined
Nh	s'abimer	dans Na	be sunk in (grief)

Nh	abonner	Nh à Nt	take out a subscription to Nt for Nh
Nh	abonner	Nh à V Nt	subscribe Nh to receive Nt
Nh	s'abonner	à Nt	subscribe to Nt
Nh	s'abonner	à Nq	install Nq (gas, electr.)
N	aborder	à, en Nc	reach (river; China)
Nh	aborder	Nb	board (ship)
Nb	aborder	Nb	collide with (ship)
Nh	aborder	Nh	accost (Mary)
Nh	aborder	Na	tackle (subject)
Nh	aborder	Nq (virage)	take (a turn)
Nh	aborder	Nb au quai	berth (ship)
N	abréger	Na	shorten Na
N	abréger	Nt	abbreviate Nt
pour	abréger		to be brief
N	abréger	Nt de Nt	shorten Nt by Nt
N	abréger	Na de Nu	shorten (life) by (year)
N	abréger	Nt en Nt	shorten Nt to Nt
Nh	abroger	Nal	repeal (law)
Nh,Ne	actionner	Ne, Nf	activate (tool, machine)
Nh	s'actionner	0	bestir -self
	actionné	par	driven by
Nh	administrer	Na, Nh	administer (business)
Nh	administrer	Nxb à Nh	administer (drug) to Nh
Nh	affréter	Nv à Nh	hire out Nv from Nh
Nh	amidonner	Nz (de Nx)	starch (shirt) with

PAT expressions: an algebra for text search

AIRI SALMINEN — FRANK W.M. TOMPA

1. Introduction

Text search operations are used to locate and retrieve needed information from some text collection. In traditional information retrieval, text search is a means for identifying relevant documents [Salton83, Lee85]. By specifying selection criteria for the text of a document, the reader can choose a subset of a given set of documents. If the text collection is defined not as a set of documents, but more generally as a structure containing some parts, then text search involves the specification of those parts of interest to the reader. The structure of the documents may be determined by the search system, by the author, by the text installer, or by the reader.

In the PAT™ system [Gonnet87a, Fawcett89a, Fawcett89b] text search operations are expressions that efficiently combine traditional search capabilities with some new, powerful facilities. PAT contains means for lexical search, proximity search, contextual search and Boolean search [Hollaar79, Larson84, Lee85, Burkowski91]. It also contains more rare operation types, including position and frequency search. Furthermore, a novel feature in PAT is the capability by which a reader can define structures for a text and use these structures in subsequent operations. One of the goals of this paper is to introduce the powerful search capabilities of PAT expressions.

Text search is usually considered so simple that only a rough description of the operations is given. For example, when word search is discussed, we are seldom told what is meant by a "word". The reader has to find out through experimentation how many words are contained in the strings "Jean-Marie" and "O'Hara". However, a careless description of search operations may lead to search errors or unnecessarily long retrieval sessions. A second goal of the paper, therefore, is to introduce a mechanism for precise specification of text search semantics.

---

† PAT is a registered trademark of Open Text Corporation.

Text search using PAT is typically simple and straightforward [Raymond90]. However, because of the powerful definition capabilities included in PAT, explaining and understanding the semantics of some operations may be difficult. As a side-effect of our systematic specification of PAT, we have identified some features of PAT expressions that cause problems and thus would benefit from further development. From this we see that precise specification also serves as a means for evaluation and offers a means for comparing text search systems.

As is common in information retrieval systems, a PAT search is applied to indexed text [see, for example, Gonnet83, Croft84, Larson84, Faloutsos85, Salton89, Burkowski91]. Indexing is usually described from the point of view of implementation, for example, by giving an algorithm for the indexing [Salton81, Salton89, Gonnet91]. However, since the way text is indexed affects search behaviour, our systematic approach to precise description must include mechanisms that accommodate indexing definition capabilities.

2. The PAT system

PAT is a text searching system developed at the University of Waterloo's Centre for the New Oxford English Dictionary and Text Research [Berg91, Tompa92], and commercially available from Open Text Corporation in Waterloo. PAT is used, for example, for searching the 570 megabyte *Oxford English Dictionary*, among other texts and text collections. As distributed, the PAT system has two different end-user interfaces: one based on a command language using PAT expressions (Figure 1), the other based on direct manipulation interaction in which the system creates PAT expressions from the point-and-click actions of the end user (Figure 2). PAT also provides an applications programmers' interface by means of which customized front ends can be written, communicating with the search engine via PAT expressions.

The text search is based on three innovative techniques: PAT indexing, region definitions, and PAT expressions. PAT accesses the original text with the aid of an associated index [Gonnet91] that maps index terms to their occurrences in the text, which we shall call *indexed elements*. The choice of which text segments to index in PAT is not fixed by the system. Instead, PAT's indexing offers the text installer the capability to define the indexed elements of the text. The indexed elements may consist of, for example, all individual characters of the text, or multicharacter words separated by delimiters. Each indexed element is considered to be the start of a "semi-infinite string" that continues to the end of the text. In processing a query, the system finds those characters that begin semi-infinite strings matching the string given by the user. The text installer defines how the matching is determined. It is as simple to find single indexed elements or their prefixes as it is to find longer phrases, consisting of several indexed elements. If all characters are defined as indexed elements, the search in an indexed text corresponds to truly full-text search (where "the" matches the second character in "other").

In response to a query, PAT returns a result set, which is either the set of *match points* or the set of *regions* satisfying a given criterion, expressed by a PAT expression. A match point is a character (starting a semi-infinite string), whereas a region is a substring of the text beginning and ending with specified characters. Regions are defined either by the text installer or by the reader. If the text is tagged (e.g., using SGML's reference syntax [Goldfarb90]), the regions with a given name can be defined to begin and end with given tags. Alternatively, any characters found by text search can be used to denote the starts and ends of regions.

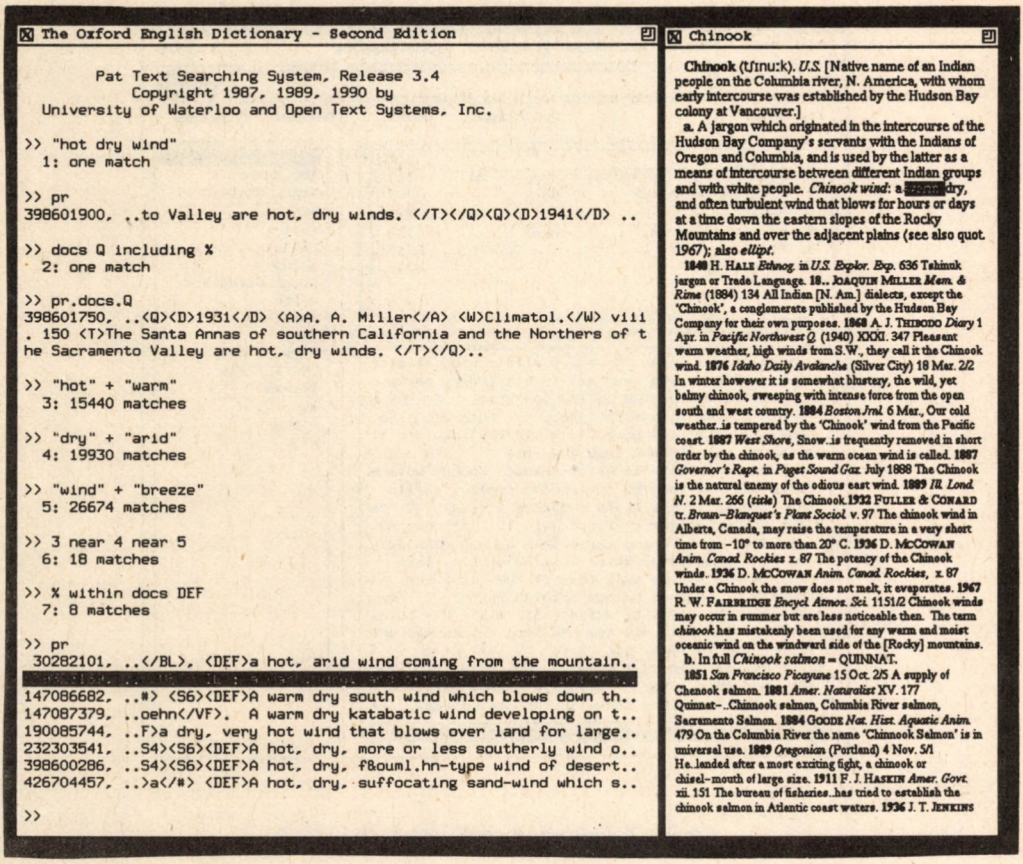


Figure 1: Command-line interface with corresponding display window

Result sets from one PAT expression can be used as operands for subsequent expressions. By default, a search command shows the count of match points or regions in the result set. If the result set is a match point set, characters from the left and right context of the match points in the set can be printed by a printing command. If the result set is a region set, the reader can print either regions from the set or characters from the left and right context of the beginning characters of the regions. In a workstation environment the text of a region may be displayed in a formatted form in a LECTOR™ window as in Figure 1 [Raymond91].

† LECTOR is a registered trademark of Open Text Corporation.

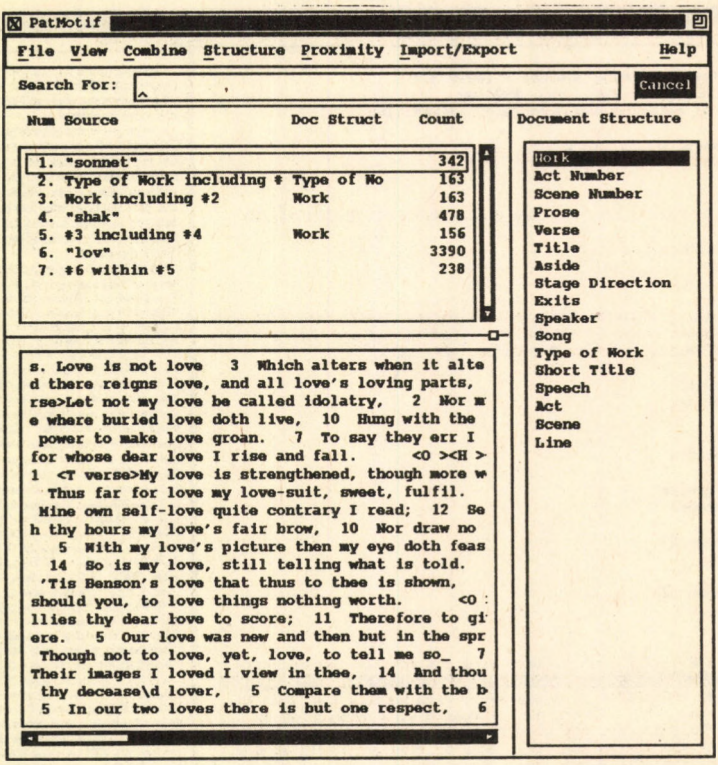


Figure 2: Point-and-click interface to PAT

3. Text in PAT

In earlier work, grammars were used to describe indexing as a view of the original text, and this view was then further used to specify text retrieval operations [Tague91]. In this paper we will use the same approach, based on the text model introduced in [Salminen92], which extends the earlier grammar-based model of [Gonnet87b].

3.1. Text as a character string

Any PAT text can be viewed as a sequence of characters. The following grammar, consisting of two productions, can be used as the schema for any such text:

(S) **string ::= character+ .**
character ::= 'a' |'b' |... .

The grammar defines two text *types*: **string** and **character**. The first production indicates that a string consists of one or more characters. (The symbol + in the production denotes iteration one or more times.) Thus, a *value* of type **string** is any nonempty sequence of values of type **character**. The second production enumerates all available values of type **character**. (The symbol | separates alternatives.)

Consider the following sample text:

```
<h>Consumer spending in U.S. up 1.5 per cent in June</h>
```

When we use this text as a text context for the schema (S), it is viewed as having been parsed by the grammar, and we can refer to *parts* of the text. Each part is a text entity corresponding to one of the defined types. The sample text contains one part of type **string**, i.e. the whole text, and 56 parts of type **character**. The value of the **string** part is the complete sample text above. The value of the first **character** part is "<", the value of the second **character** part is "h", the value of the 49th **character** part is "J", and so on. PAT capitalizes on the fact that each **character** part is identified with a unique position in the whole text.

3.2. Text as an indexed string

Alternatively, PAT text can be viewed as a string consisting of indexed elements and delimiters separating indexed elements. Each indexed element begins a new phrase, continuing to the right from the indexed element until the end of the text. Therefore, if the string contains *n* indexed elements, it also contains *n* such phrases. For the above sample text, if (for simplicity) each contiguous sequence of non-blank characters is an indexed element and blanks are delimiters, there are ten phrases corresponding to the ten indexed elements:

```
<h>Consumer spending in U.S. up 1.5 per cent in June</h>
spending in U.S. up 1.5 per cent in June</h>
in U.S. up 1.5 per cent in June</h>
U.S. up 1.5 per cent in June</h>
up 1.5 per cent in June</h>
1.5 per cent in June</h>
per cent in June</h>
cent in June</h>
in June</h>
June</h>
```

We can define this formally using the following rules:

(a) string ::= [delimiter] phrase .
(b) phrase ::= indexed_element [delimiter] [phrase] .

The first production indicates that a string may be a phrase or it consists of a delimiter followed by a phrase. (The square brackets denote optionality.) The second production shows that a phrase always begins with an indexed element, which may be followed by a delimiter and then by another phrase. Thus, two indexed elements may be separated by a delimiter,

or one indexed element may follow another directly.

Indexed elements and delimiters are described specifically for each text file by the text installer. For most text retrieval applications, text installers choose indexed elements to correspond as closely as possible to the users' perception of "words". Figure 3 shows a possible grammar describing the indexing of a text containing standard markup tags (e.g., this grammar describes the indexing used for the *Oxford English Dictionary* at the University of Waterloo and elsewhere).

```
(1)   string ::= [delimiter] phrase .
(2)   phrase ::= indexed_element [delimiter]
                 [phrase {where indexed_element {is preemptive_element or preceded by delimiter} }] .
(3)   indexed_element ::= preemptive_element | limited_element .

(4)   preemptive_element ::= stand_alone_char | signal_char element_char* .
(5)   limited_element ::= element_char+ .
(6)   delimiter ::= delim_char+ .

(7)   stand_alone_char ::= hyphen .
(8)   signal_char ::= less | ampersand .
(9)   element_char ::= a | ... | z | A | ... | Z | d1 | ... | d9 | hash | slash .
(10)  delim_char ::= blank | period | comma | greater | colon | apostrophe | ... .

(11)  hyphen ::= '-' .
      less ::= '<' .                                  ampersand ::= '&' .
      a ::= 'a' .              ...                    z ::= 'z' .
      A ::= 'A' .              ...                    Z ::= 'Z' .
      d0 ::= '0' .             ...                    d9 ::= '9' .
      hash ::= '#' .           slash ::= '/' .
      blank ::= ' ' .          period ::= '.' .       comma ::= ',' .
      greater ::= '>' .        colon ::= ':' .        apostrophe ::= ''' . ... .
```

Figure 3: A grammar describing an indexed text.

The grammar defines one way to divide a string into indexed elements and delimiters. So that an indexed element can be easily recognized syntactically, it must either begin with a specific character (**preemptive_element**) or it must be preceded by a delimiter. This constraint is specified in production (2) by replacing the use of the simple non-terminal in (b) above by the "property" **phrase {where indexed_element {is preemptive_element or preceded by delimiter}** } (see [Salminen92] for a detailed explanation of properties). Preemptive elements are either single characters (for this specific grammar, hyphens constitute the only such elements), or they start with a **signal_char** (for this grammar, either a less-than-sign or an ampersand) and then continue with zero or more element characters (here any combination of letters, digits, hash-sign, or slash). Delimiters contain characters distinct from any of these characters. Productions (7) - (10) contain the text-specific assignment of characters to the four classes that define which substrings of the text are indexed elements. They use the nonterminals defined in productions starting from (11) so that character transformations can be separately specified for evaluating whether a given phrase matches a query (see Section 4.1). The effect of this particular choice of indexing is that most punctuation characters are treated as blanks (periods, commas, etc. are not part of any indexed elements).

Consider again the sample text above. As PAT text this is a sequence of characters, containing parts of types **string** and **character**. However, using Figure 3 as an indexing description for the text means that the text is reparsed by the grammar. The text context then includes simultaneously parts having types from the schema (S) and parts having types from the indexing description. The text context contains 14 indexed elements which are shown below (the extent of each marked by |-----).

```
<h>Consumer spending in U.S. up 1.5 per cent in June</h>
|- |------- |------- |- | |  |- | | |-- |--- |- |---|--
```

Notice that delimiters are not contained in **indexed_element** parts. Furthermore, the two last indexed elements are not separated by a delimiter: the character '<' begins a new indexed element immediately.

The grammar described in Figure 3 has been used most often for PAT applications. From the grammar it is clear that "Jean-Marie" contains three indexed elements and "O'Hara" contains two. An alternative index that makes *every* character an indexed element has been used to support detailed proofreading of the *Oxford English Dictionary* in preparation for the second edition. Between these two extremes exist many other possibilities: for example, specification and program text can be indexed on all upper case letters as well as conventional word starts so as to allow a software engineer to find all mention of functions and variables having to do with a "window" even if they are named "AdjustWindowImage" and so forth. Such indexing can be simply described through reassignment of upper case letters to **signal_char** instead of **element_char**.

This is not intended to describe the way PAT indexing is implemented [Gonnet91]. Instead, we describe the indexing from a reader's viewpoint, specifically, the effect of indexing on search operations. From Figure 3, the addressable units of a text are clearly identified. As opposed to the simplistic listing of phrases given above, it is precisely stated that the sample text with such an index has 14 retrievable phrases:

```
<h>Consumer spending in U.S. up 1.5 per cent in June</h>
Consumer spending in U.S. up 1.5 per cent in June</h>
spending in U.S. up 1.5 per cent in June</h>
in U.S. up 1.5 per cent in June</h>
U.S. up 1.5 per cent in June</h>
S. up 1.5 per cent in June</h>
up 1.5 per cent in June</h>
1.5 per cent in June</h>
5 per cent in June</h>
per cent in June</h>
cent in June</h>
in June</h>
June</h>
</h>
```

Because the indexing described in Figure 3 includes period as a delimiter character, the strings "U.S." and "1.5" each contain two indexed elements. The current PAT indexing capability is very simple, and we cannot define context-dependent differences in the use of a character. Thus even though the strings "U.S." and "1.5" might seem more appropriately considered as one indexed element each, we cannot define them so without causing end-of-sentence periods to also be included in indexed elements.

Slashes may also cause some problems with this indexing. Because the indexing description is made for text with standard markup begin and end tags (for example, "<h>" and "</h>"), it is defined such that each begin and end tag contains one indexed element: for example, "<h" and "</h". (The character '>' ending a tag is defined to be a delimiter.) Slash must be defined as an indexed element character if the identifier within the end tag (e.g., the "h" within "</h>") is not to be made a separate indexed element. However, as a result the substring "USA/Canada" within a text context will also be treated as one indexed element. Subsequently, if the reader searches for occurrences of "Canada", the substring within "USA/Canada border" fails to match.

From these examples, we see that defining the indexing of text properly is somewhat problematic with PAT. By extending PAT's indexing definition capabilities, more flexibility for application-dependent indexing could be achieved. However, whatever the indexing definition techniques, the richness of natural language and the variety of information needs from natural language text will always cause problems in defining indexing satisfactorily.

3.3. Regions

A region is a substring of PAT text, beginning and ending at specified characters. Each region belongs to one or more *region sets*. A region cannot overlap other regions belonging to the same region set, but it can overlap regions in other sets arbitrarily. This concept is thus a unification and generalization of the concepts of "document" and "field" used in conventional information retrieveal systems.

From within PAT, a region set can be created by a region definition, which gives the condition determining the first and last characters of each component region. Region sets can also be derived from prior region sets. The form of the region definitions is described in Section 4.4.

A region definition corresponds to a new grammar by which the text context can be reparsed, and through which new parts can be identified in the text. For example, for the following text we could define a region set, named "year", to include the two regions as indicated:

```
Fascicles of the OED appeared between 1884 and 1928.
                                    ----     ----
```

The capability to handle regions is the feature that most distinguishes PAT from conventional document retrieval systems. Regions are defined either by the text installer (predefined regions) or by the reader (user-defined regions), the latter serving as temporary definitions of scope or as personal "views" of the text for limiting queries or responses. It is primarily through judicious definition of region sets that texts can be structured to meet the

needs of diverse applications.

4. PAT operations

PAT is a set-at-a-time algebra for manipulating results of text queries. Each PAT expression is either a match point expression, specifying a set of characters in the text context, or a region expression, specifying a set of regions in the text context. The sets specified by match point expressions or region expressions are called match point sets and region sets, respectively; collectively they are called *result sets*.

Result sets produced by search commands are numbered sequentially, and they can be referenced in subsequent expressions by number. A user can optionally assign a name to a result set via a search command, in which case the result set can be subsequently referenced by name. If *e* is a PAT expression, then a command of the form

$$n = e$$

gives the name **n* to the result set specified by *e*. If *e* is a match point expression, then **n* refers to the corresponding match point set; if *e* is a region expression, **n* names the corresponding region set. Finally, the symbol % refers to the immediately preceding result set.

The syntax of PAT expressions is described in the Appendix. PAT operations can be classified by type:

	search class	*result set type*
(1)	lexical search	match points
(2)	position search	match points
(3)	frequency search	match points
(4)	region definition	regions
(5)	restriction	match points / regions
(6)	augmentation	match points / regions

As indicated in the table, the resulting expressions in a class can be match point expressions or region expressions only, or both.

Expressions in classes (1), (2), and (3) are always match point expressions, i.e. they are used to search for characters. In lexical search the user searches for phrases by giving one or two patterns. The result set consists of the first characters in the found phrases. Position search means searching for a character in a given position in the whole text, or searching for characters at fixed offsets to the left or right of the match points of a given match point set. Frequency search means identifying match points for frequently appearing substrings or long repetitions from the text. Expressions from classes (1), (2), and (3) are discussed further in Sections 4.1, 4.2 and, 4.3, respectively.

Expressions in class (4) are always region expressions: a region definition specifies a region set as a function of two match point sets. Region definitions are discussed further in Section 4.4.

Expressions in classes (5) and (6) are either match point expressions or region expressions. These produce new result sets as a function of existing sets and are discussed in more detail in Sections 4.5 and 4.6, respectively.

4.1. Lexical search

A PAT expression for lexical search is either a character string or of the form $s_1..s_2$ where s_1 and s_2 are character strings. Lexical search always yields a match point set.

By giving a string s the reader searches for all characters in the text that begin phrases matching s. The matching of a phrase with a string pattern is determined after normalizing both the phrase and the pattern. The normalization inherent in PAT maps delimiter characters to blanks. However, concurrently with the indexing, the text installer may describe additional normalization, through which characters can be deleted or replaced by alternative characters.

In grammar based text modelling, normalization can be described as a text translation, defined by a set of translation productions that redefine those text types of the indexing description whose parts are changed by normalization. The translation productions, together with the indexing description, define a translation of a character string to a normalized string.

Figure 3 showed one possible indexing for a text. For such a text, normalization could be defined by a table showing replacement productions as follows:

indexing production	*normalization replacement*
string ::= [delimiter] phrase .	**string ::= phrase .**
delimiter ::= delim_char+ .	**delimiter ::= ' ' .**
A ::= 'A' .	**A ::= 'a' .**
...	...
Z ::= 'Z' .	**Z ::= 'z' .**

The replacement for the **string**-production indicates that if the string begins with a delimiter, the delimiter is removed in normalization. The **delimiter**-production causes the replacement of all delimiters by blanks, and the rest of the productions define that each upper case letter is replaced by the corresponding lower case letter. The effect of these replacements is that pattern matching is case-insensitive, delimiters cannot be distinguished, and the presence or absence of a delimiter before the start of a match cannot be determined.

As described in Section 3.2, PAT allows a text installer to define the partitioning of characters into the classes **stand_alone_char**, **signal_char**, **element_char** and **delim_char** to customize indexing. The text installer customizes normalization by assigning replacement characters as well, under the constraint that replacements must be chosen from the same class as the character being replaced (e.g., lower case letters can be substituted for upper case letters only if both are in the same class, in this case **element_char**). There is also a limited facility for defining stopwords, that is, specific characters strings that are to be treated as delimiters rather than as indexed elements. Extensions to the language are being developed to generalize these capabilities by allowing a text installer to define the indexing and normalization transductions using a powerful language for describing Mealy machines [Gonnet92]. In the examples of the rest of the paper we assume that the indexing is defined by the grammar in

Figure 3 and the normalization by the translation productions given above.

A phrase in the text matches a given string if the normalized string is a prefix of the normalized phrase. Consider our earlier sample text:

```
<h>Consumer spending in U.S. up 1.5 per cent in June</h>
```

The string "in" matches two phrases:

```
in U.S. up 1.5 per cent in June</h>
in June</h>
```

The string "s" also matches two phrases:

```
spending in U.S. up 1.5 per cent in June</h>
S. up 1.5 per cent in June</h>
```

because through normalization the sample text is translated to the form

```
<h consumer spending in u s up 1 5 per cent in june</h
```

Thus for each of these lexical searches, the expression denotes a result set consisting of two match points: the first characters of each matching phrase. Note that the strings "U.S. ", "u s u", and "..., 'U:'" all match the phrase "U.S. up 1.5 per cent in June</h>":

string	*normalized form*
U.S. up 1.5 per cent ...	u s up 1 5 per cent ...
U.S.	u s
u s u	u s u
.., 'U:'	u

By giving two strings $s_1..s_2$ the reader searches for all characters that begin phrases such that the normalized phrase matches either normalized s_1 or normalized s_2 as above or it follows normalized s_1 and precedes normalized s_2 in lexicographic order. In many applications this kind of search capability is very useful and may often replace a long sequence of searches for one string at a time [Logan88]. As an example consider the text

```
shortages hit in 1973 and 1979. In the 1980s ... in 1978 ...
```

For the expression "hi.."jo", the result set contains four match points corresponding to the initial characters of the phrases

```
hit in 1973 and 1979. In the 1980s ... in 1978 ...
in 1973 and 1979. In the 1980s ... in 1978 ...
In the 1980s ... in 1978 ...
in 1978 ...
```

which can be represented diagrammatically by the notation

```
shortages hit in 1973 and 1979. In the 1980s ... in 1978 ...
          |   |                 |               |
```

For the expression "1975".."1980" the result set contains three match points:

```
shortages hit in 1973 and 1979. In the 1980s ... in 1978 ...
                          |           |           |
```

4.2. Position search

Position search is used to find a character in a given position in the whole text, or characters at a given distance to the left or right of match points in a match point set. The expression for the first kind of search is

[*n*]

where *n* is a positive integer. As an example, with respect to the following text the PAT expression "[15]" denotes the 15th character, namely the "u" in "Yugoslavs", and thus the indicated match point:

```
Some 700,000 Yugoslavs live in Germany, most of them Croats.
             |
```

Unlike for lexical searches, the match point denoted by a position search need not be the initial character of an indexed element.

Match points can be shifted by expressions of the form

shift.*n e*

where *n* is a positive or negative integer and *e* is an expression. The expression *e* is considered to specify a match point set: if it is a region expression, the match points correspond to the first characters of the regions. For example, with the expression

shift.3 "1800".."2000"

we can find the two indicated match points in the following text:

```
Fascicles of the OED appeared between 1884 and 1928.
                                    |        |
```

Again the match points in the result set need not correspond to starts of phrases.

4.3. Frequency search

PAT includes two groups of frequency expressions, used to find frequently occurring or repeated substrings in a text. The "signif" expressions are used to find the most frequent substrings, consisting of whole indexed elements, and beginning a phrase that matches a given string. The "lrep" expressions are used to find the longest repeated substrings, each consisting of whole indexed elements, beginning with a phrase that matches a given string. The frequency search expressions are always match point expressions.

Consider the frequency expression

signif *e*

where *e* is a match point expression. This operation first normalizes the character strings starting at match points corresponding to *e* and identifies for each one its prefix up to the first delimiter. From the corresponding match points, "signif" returns the subset associated with the most frequently occurring prefix. In the complete works of Shakespeare [OUP88], 792 words begin with the string "thro". The command

```
signif "thro"
```

returns a match point set with 329 members, corresponding to the initial characters of each occurrence of the word "through" in the text, that being the most frequent word beginning with "thro".

Using the form

signif.*n e*

a user can specify that extended prefixes, including *n* delimiters rather than stopping at the first one, should be compared. For Shakespeare, the following results can be obtained:

expression	*number of matches*	*matching phrase*
signif.2 "thro"	107	`through the`
signif.3 "thro"	8	`through the world`
signif.4 "thro"	2	`throat our height can`

Using the form

signif.–*n s*

the user can retrieve result sets corresponding to the *n* most frequent normalized prefixes beginning with string *s*. For Shakespeare, the following example illustrates this form:

```
signif.-10 "thro"

  329 matches, text= through
  116 matches, text= throw
  107 matches, text= through the
   77 matches, text= throne
   58 matches, text= throat
   46 matches, text= throws
   35 matches, text= thrown
   31 matches, text= throats
   21 matches, text= throwing
   20 matches, text= throng
```

Note that since the two word prefix "through the" occurs more frequently than do remaining single words, the results set corresponding to this extension of "through" is returned before, for example, the match points corresponding to "throne". Because a *sequence* of match point sets are returned, this form of the command cannot be used within other PAT expressions.

The operation

lrep *e*

chooses from the match points identified by *e* those having the longest normalized extensions such that there are at least two occurrences of the same extension. The result set consists of the subset of match points corresponding to the first characters of these phrases. Using the form

lrep.*n e*

all repeated phrases having at least *n* characters in the common prefix can be identified. Looking again at Shakespeare,

```
lrep "thro"
```

returns a result set containing two match points corresponding to a repeated phrase having 162 characters (including line numbers from the play):

```
                ..throw incense. Have I caught thee?
22  He that parts us shall bring a brand from heaven
23  And fire us hence like foxes. Wipe thine eyes.
24  The goodyear shall devour ['em, flesh and fell,]

                ..throw incense. Have I caught thee?
22  He that parts us shall bring a brand from heaven
23  And fire us hence like foxes. Wipe thine eyes.
24  The goodyear shall devour [them, flesh and fell,]
```

which represents a variant edition included in the text. Similarly,

lrep.100 "thro"

returns a result set with four match points, two of which are as above and the other two representing another variant edition:

```
                ..through itself to that full issue
4  For which I razed my likeness. Now, banished Kent,
5  If thou canst serve where thou dost stand condemned,
6  [So may it come thy master, whom thou lov'st,]

                ..through itself to that full issue
4  For which I razed my likeness. Now, banished Kent,
5  If thou canst serve where thou dost stand condemned,
6  [Thy master, whom thou lov'st...]
```

Frequency expressions have been included in PAT for the needs of linguists and editors. In an experiment with users the frequency search operations were found among the most difficult to use [Raymond90]. Therefore, improvements resulting in more useful frequency search operations are currently being investigated.

4.4. Region definitions

A region definition specifies a region set consisting of new regions. It has the form

$$\text{docs } e_1..e_2$$

where e_1 and e_2 are match point expressions. Expression e_1 gives the condition for the first character of a new region and e_2 a condition for the last character. If e_1 or e_2 is a region expression, it denotes the set of the first characters of the argument regions.

Suppose we define

year = docs ("1800 .."2000 ") .. (shift.3 "1800 .."2000 ")

Then both years in our earlier sample text would be regions in the set named by the expression *year:

```
Fascicles of the OED appeared between 1884 and 1928.
                                      ----     ----
```

Alternatively, if a region set named "year" had been defined by the text installer, it could be referenced by the expression "docs year".

From a tagged text we can define regions by matching the tags. For example, if the text consists of articles with headlines, publication dates, authors, and paragraphs, such that each of these parts is denoted by tags, we can define the structure in terms of PAT expressions. Specifically, assuming that headlines are denoted by the tags "<h>" and "</h>", the definition

```
headline = docs "<h>"..(shift.3 "</h>")
```

defines a region set called *headline in which each element spans the substring from the first character of the opening tag to the last character of the closing tag.

Each region definition creates a region set independently of other region definitions. There are no constraints on how regions in one set overlap earlier defined regions. However, the regions defined in one definition are not self-overlapping. Thus, if a text includes the following fragment:

```
<h>Editor denies <h>Massive Failure</h> misleading</h>...
```

the definition of headline given above would include the region corresponding to the substring "<h>Massive Failure</h>" but not the surrounding headline.

4.5. Restriction

Users often require means to select subsets from a given set. PAT provides several binary operators of the form

$$e_1 \; op \; e_2$$

which designate result sets that are subsets of the set corresponding to e_1, consisting of elements that satisfy the condition expressed by "*op* e_2". Thus if e_1 is a region expression, the result is a region set; if e_1 is a match point expression, the result is a match point set.

The first operator "including" allows users to find those regions that contain at least one member of a given match point set. For example, consider again the complete works of Shakespeare and assume that the name *speech has been defined to designate the set of regions corresponding to all speeches from all the plays. Hence, the expression

```
*speech including "wherefore art"
```

returns the subset of speeches containing match points corresponding to the lexical search, namely, the two speeches:

```
 ..<S JULIET> <T asd> {(not knowing Romeo hears her)}<T verse> O Romeo,      +
75  Romeo, wherefore art thou Romeo?
76  Deny thy father and refuse thy name,
77  Or if thou wilt not, be but sworn my love,
78  And I'll no longer be a Capulet.
```

and

```
    ..<S FLAVIUS><T verse> But wherefore art not in thy shop today?
28  Why dost thou lead these men about the streets?
```

In general, the region expression

$$e_1 \text{ including.}n\ e_2$$

yields a result set consisting of those regions in the set specified by e_1 which contain at least n match points specified in e_2 (with ". 1" as the default). Thus

*speech including.7 "Romeo"

returns one speech (Juliet's final monologue).

Similarly,

$$e_1 \text{ not including.}n\ e_2$$

returns the complementary subset. Thus, to find all Shakespeare's speeches that include the word "dream" but no word beginning with "sleep", a user can write the expression

(*speech including "dream ") not including "sleep"

If the constraining expression is a region expression, then the match point set used to test membership in the result set consists of the first characters of the regions. Continuing with the previous examples, if *scene designates all scenes from Shakespeare's plays,

*scene including.100 *speech

returns all scenes having 100 or more speeches. However, it is important to recognize that PAT only checks that the *start* of the speech is in the scene, which for nested regions is sufficient for the *whole* speech to be in the scene. Thus it follows that if "line" designates all lines from the plays,

*line including *speech

returns the set of lines that contain starts of speeches, as opposed to those containing complete speeches.

The "including" operator produces a result set that is a subset of a region set. PAT provides other operators that produce a subset of an *arbitrary* set by restricting membership according to the following constraints:

op	*explanation*
^	coincident with some member of e_2
—	not coincident with any member of e_2
fby.n	preceding some member of e_2 by at most n characters
near.n	separated by at most n characters from some member of e_2
within	contained in some region designated by e_2

A user wishing to know which sonnets contain suffixed instances of the word "love" could write the expression

*sonnet including ("lov" – "love ")

assuming prior construction of the set *sonnet. The expression "lov" includes (among many others) match points in the following lines from several sonnets:

```
13  And so of you, beauteous and lovely youth,
 3  Both grace and faults are loved of more and less;
14  Those that can see thou lov'st, and I am blind.
 3  Have put on black, and loving mourners be,
 3  But 'tis my heart that loves what they despise,
13  So true a fool is love that in your will,
 9  Rise, resty muse, my love's sweet face survey
```

the last two of which are also included in the set denoted by the expression "love " and therefore excluded from the difference set.

For all of these operators, the set constraints are based on match points. Thus if a region expression is involved, the constraint on each member is evaluated in terms of the match point corresponding to its first character. For example,

*sonnet fby.100 "love "

returns regions in the set *sonnet for which the word "love" occurs within the first 100 characters; the proximity is measured from the start of the region, not the end. This semantics occasionally causes misunderstandings and errors on the part of some users and should therefore be reconsidered.

In an expression of the form

$$e_1 \text{ within } e_2$$

e_2 must be a region expression. For example,

("lov" – "love ") within *sonnet

returns the match points corresponding to suffixed uses of "love" occurring in sonnets (contrast with the use of "including" above), and

sonnetline = *line within *sonnet

returns the regions corresponding to lines within sonnets. As for the previous operators, when e_1 is a region expression, the constraint is based on the match points corresponding to the starts of the regions, but the result set is a (region) subset of e_1.

4.6. Augmentation

Because PAT provides restriction operations corresponding to set intersections "^" and set difference "–", it also includes the binary operator "+" to indicate set union. As expected, when the two operands are match point expressions, the result set is a match point set including all members of both argument sets. For example, to find all sonnet lines that include the words "boy" or "youth", one can write

BoyOrYouth = *sonnetline including ("boy " + "youth ")

which will return 18 lines. To find in which sonnets such lines appear (assuming prior definition of the region expression *title), a user could write

*title within (*sonnet including *BoyOrYouth)

yielding 16 regions containing the titles. To examine these titles together with the sonnet lines, one could then take the union of the two region sets:

*BoyOrYouth + %

(where % refers to the previous result), yielding the following text:

```
[[Sonnet]] 2
  3  Thy youth's proud livery, so gazed on now,
[[Sonnet]] 7
  6  Resembling strong youth in his middle age,
[[Sonnet]] 11
  4  Thou mayst call thine when thou from youth convertest.
[[Sonnet]] 15
 10  Sets you most rich in youth before my sight,
 12  To change your day of youth to sullied night;
[[Sonnet]] 22
  2  So long as youth and thou are of one date;
[[Sonnet]] 37
  2  To see his active child do deeds of youth,
[[Sonnet]] 41
 10  And chide thy beauty and thy straying youth
[[Sonnet]] 54
 13  And so of you, beauteous and lovely youth,
[[Sonnet]] 60
  9  Time doth transfix the flourish set on youth,
[[Sonnet]] 73
 10  That on the ashes of his youth doth lie
[[Sonnet]] 96
  1  <T verse>Some say thy fault is youth, some wantonness;
  2  Some say thy grace is youth and gentle sport.
[[Sonnet]] 98
  3  Hath put a spirit of youth in everything,
[[Sonnet]] 108
  5  Nothing, sweet boy; but yet like prayers divine
[[Sonnet]] 110
  7  These blenches gave my heart another youth,
[[Sonnet]] 138
  3  That she might think me some untutored youth
[[Sonnet]] 153
 10  The boy for trial needs would touch my breast.
```

It should be noted that the semantics of PAT expressions are different from those of Boolean operators in traditional document retrieval systems. A naïve user wishing to find sonnets that include the word "love" as well as either "boy" or "youth" might write

```
*sonnet including (("boy " + "youth ") ^ "love")
```

and be surprised when PAT reports "no match". The semantics of PAT expressions indicate that the correct way to pose this query is

```
(*sonnet including ("boy " + "youth ")) including "love"
```

If one of the operands of the union operator is a match point expression and the other is a region expression, a simple union of the sets cannot be performed; PAT instead defines the result to be a match point set, using the match points corresponding to the starts of the argument regions in place of the regions themselves. Similarly, if both arguments are region expressions, but the regions in the corresponding sets overlap, PAT cannot form a simple union, since overlapping regions in one set are disallowed. Thus in this situation, PAT again defines the result to be a match point set, using the match points corresponding to the starts of the regions in place of the regions themselves. As a result,

```
(*speech including "to be or not to be") + (*line including "dying")
```

is a *region* expression denoting 58 regions of text, but

```
(*speech including "to be or not to be") + (*line including "death")
```

is a *match point* expression denoting 1030 match points in the text, since two lines containing the word "death" occur within the speech. Although these semantics are self-consistent, this feature in PAT is easily misunderstood and therefore changing it should be considered, perhaps by suitably defining the union of overlapping regions (as done, for example, in [Burkowski91]).

5. Conclusions

We have described the query capabilities included in the PAT text search system. We have divided PAT expressions into six classes and introduced the syntax and semantics of the expressions in the classes. We have shown that PAT indexing can be specified by productions as a view of PAT text seen as a character sequence. The matching of a phrase with a given string was also described by productions and text translation.

During the specification of PAT's search capabilities we have identified some problem areas which are interesting, not only from the point of view of PAT evaluation, but also in the evaluation of search capabilities in text search systems more generally. The indexing definition capabilities in PAT offer a means for application dependent indexing. However, the current indexing definition techniques are limited and designed for homogeneous text, whereas some document collections, such as multilingual collections, might include texts with heterogeneous indexing needs. Open Text Corporation supports a facility known as "Parallel PAT" that provides a single PAT interface to a collections of independently managed PAT texts, each using its own indexing definition.

The flexible definition of regions is an important and original feature in PAT. Together with the indexing definition capability, this feature supports application dependent handling of text. Regions may be defined both during the text installation phase and by the reader.

Using PAT expressions the reader may search for either match points or regions, using a uniform syntax and semantics. In most cases the user may consider the restriction operations as if they were truly region operations, but occasionally the semantic implications are surprising. This is even more apparent in the augmentation operation. To support users familiar with more conventional document-based Boolean searching, a variety of front ends should be designed and implemented using PAT as a back-end search engine invoked through the program interface using PAT expressions. To date, experience with a restricted front end that provides simple look-ups in the *Oxford English Dictionary* and with a graphics-based front end to search an automobile engine manual, among others, shows that the separation of end-user convenience from search capabilities can be successful.

PAT with its region definition capabilities is designed for handling structured text. The structure of text is often expressed by tags, which can be used to define most regions. After the region definitions, the user might want to handle the text in the form where all tags are hidden. In the workstation environment, LECTOR allows a user to read regions without reading tags. For example, the reader of the *Oxford English Dictionary*, Shakespeare, or the Bible can read text displayed in a form similar to that in the printed volumes. However, the search is always applied to the tagged text, requiring users to be aware of the tags. Through the introduction of regular expressions in the definition of PAT indexing and search, more convenient search can be supported, wherein, for example, standard markup tags can be defined to be delimiters. The development of text search systems where the user performs *all* operations on text in various forms is an area of ongoing research.

Finally, the unique PAT indexing technique has made it possible to implement frequency search operations. However, the semantics of the current frequency operations is difficult to define, and they are limited in utility. Thus further studies for the development of these operations are needed.

Acknowledgements

Gaston Gonnet was the principal designer and implementer of PAT; several others, including Tim Snider and Byron Weber Becker, contributed substantially to the implementation. Financial support from the Natural Sciences and Engineering Research Council of Canada, under grants CRD0000862 and STR0045480, and from the Academy of Finland is gratefully acknowledged.

References

[Berg91] Berg, D.L., Gonnet, G.H., and Tompa, F.W., The New Oxford English Dictionary Project at the University of Waterloo, in *Computational Lexicology and Lexicography: Special Issue Dedicated to Bernard Quemada* (edited by A. Zampolli, L. Cignoni, and C. Peters), (series: Linguistica Computazionale, Vol. VII), Giardini Editori, Pisa, 1991, 29-44.

[Burkowski91] Burkowski, F.J., Textriever: A retrieval engine for multimedia databases, *Proc. of the Int. Conf. on Multimedia Information Systems*, Singapore 1991, 71-76.

[Croft84] Croft, W.B., Implementing a text storage and retrieval tool for the office, *Proc. of the First Int. Conf. on Office Automation*, IEEE Computer Society, 1984, 137-144.

[Fawcett89a] Fawcett, H., PAT3.3 User's Guide, UW Centre for the New OED and Text Research, University of Waterloo, 1989.

[Fawcett89b] Fawcett, H., PAT Installation Guide, UW Centre for the New OED and Text Research, University of Waterloo, 1989.

[Faloutsos85] Faloutsos, C., Signature files: Design and performance comparison of some signature extraction methods, *Proc. of ACM-SIGMOD 1985, Int. Conf. on Management of Data, SIGMOD Record 14*, 4 (Dec. 1985), 63-82.

[Goldfarb90] Goldfarb, C.F., *The SGML Handbook*, Oxford University Press, 1990.

[Gonnet83] Gonnet, G.H., Unstructured databases — or — very efficient text searching, *Proc. of ACM Principles of Database Systems*, 1983.

[Gonnet87a] Gonnet, G.H, Examples of PAT applied to the *Oxford English Dictionary*, Tech. Rept. OED-87-02, UW Centre for the New OED and Text Research, University of Waterloo, 1987.

[Gonnet87b] Gonnet, G.H. and Tompa, F.W., Mind your grammar: a new approach to modelling text, *Proc. of the 13th Int. Conf. on Very Large Data Bases*, 1987, 339-346.

[Gonnet91] Gonnet, G.H., Baeza-Yates, R.A., and Snider, T., Lexicographical indices for text: inverted files vs. PAT trees, Tech. Rept. OED-91-01, UW Centre for the New OED and Text Research, University of Waterloo, 1991.

[Gonnet92] Gonnet, G.H. and Snider, T., Mealy Machines, unpublished manuscript, UW Centre for the New OED and Text Research, University of Waterloo, 1992.

[Hollaar79] Hollaar, L.A., Text retrieval computers, *Computer 12*, 3 (March 1979), 40-50.

[Larson84] Larson, P.-A., A method for speeding up text retrieval, *ACM Data Base 15*, 2 (Winter 1984), 19-23.

[Lee85] Lee, D.L., The design and evaluation of a text-retrieval machine for large databases, Ph.D. Thesis, Computer Systems Research Institute, University of Toronto, 1985.

[Logan88] Logan, H.M. and Logan, G. An inquiry into inquiry systems: a discussion of some applications of data retrieval to the New OED database, *Proc. of the Fourth Conf. of the UW Centre for the New OED and Text Research*, "Information in Text," Waterloo, ON, 26-28 October, 1988, 81-95.

[OUP88] Oxford University Press, *Complete Electronic Shakespeare*, 1988.

[Raymond90] Raymond, D.R. and Fawcett, H.J., Playing detective with full text searching software, *Proc. of SIGDOC '90, SIGDOC Asterisk 14*, 4 (1990), 157-166.

[Raymond91] Raymond, D.R, Flexible Text Display with LECTOR, *Computer 25*,8 (August 1992).

[Salminen92] Salminen, A. and Tompa, F.W., Data modelling with grammars, unpublished manuscript, UW Centre for the New OED and Text Research, University of Waterloo, 1992.

[Salton81] Salton, G., A blueprint for automatic indexing, *ACM SIGIR Forum 16*, 2 (Fall 1981), 22-38.

[Salton83] Salton, G. and McGill, M.J., *Introduction to Modern Information Retrieval*, McGraw-Hill, New York, 1983.

[Salton89] Salton, G., *Automatic Text Processing*, Addison-Wesley, Reading, 1989.

[Tague91] Tague, J., Salminen, A., and McClellan, C., A complete model for information retrieval systems, *Proc. of the 14th Int. ACM/SIGIR Conf. on Research and Development in Information Retrieval*, 1991, 14-20.

[Tompa92] Tompa, F.W., An Overview of Waterloo's Database Software for the *OED*, *Proc. Symp. on Historical Dictionary Databases and Data Retrieval Requirements* (edited by T.R. Wooldrich) (Toronto, October 1991), in *CCH (Centre for Computing in the Humanities) Working Papers 2* (1992) 123-143.

APPENDIX. The syntax of PAT expressions

```
PAT_expression ::=
      [match_point_name '='] match_point_expr  |
      [region_name '='] region_expr  |
      match_point_expr_sequence .

match_point_expr ::=
      lexical_search  |
      position_search  |
      frequency_search  |
      match_point_restriction  |
      match_point_augmentation  |
      '*' match_point_set_name  |
      match_point_set_number  |
      '%'  |
      '(' match_point_expr ')'  |
      region_expr .

lexical_search ::=
      string  |
      string '..' string  .

position_search ::=
      '[' positive_integer ']'  |
      'shift'  '.' integer match_point_expr .

frequency_search ::=
      'signif' [  ['.' positive_integer ] match_point_expr ]  |
      'lrep' [  ['.' positive_integer ] match_point_expr ] .

match_point_expr_sequence ::=
      'signif'  '.' negative_integer string .

match_point_restriction ::=
      match_point_expr '^' match_point_expr  |
      match_point_expr '-' match_point_expr  |
      match_point_expr [ 'not' ] 'fby' [ '.' positive_integer ] match_point_expr  |
      match_point_expr [ 'not' ] 'near' [ '.' positive_integer ] match_point_expr  |
      match_point_expr 'within' region_expr .

match_point_augmentation ::=
      match_point_expr '+' match_point_expr .

region_expr ::=
      region_definition  |
```

```
        'docs' installed_region_definition |
        region_restriction |
        region_augmentation |
        '*' region_set_name |
        region_set_number |
        '%' |
        '(' region_expr ')' .

region_definition ::=
        'docs' match_point_expr '..' match_point_expr .

region_restriction ::=
        region_expr [ 'not' ] 'including' [ '.' positive_integer ] match_point_expr |
        region_expr '^' match_point_expr |
        region_expr '-' match_point_expr |
        region_expr [ 'not' ] 'fby' [ '.' positive_integer ] match_point_expr |
        region_expr [ 'not' ] 'near' [ '.' positive_integer ] match_point_expr |
        region_expr [ 'not' ] 'within' region_expr .

region_augmentation ::=
        region_expr '+' region_expr .
```

N.B. An expression of the form

```
region_expr '+' region_expr
```

is a match point expression if some region specified by the first region expression overlaps some region specified by the other region expression.

Interaction between Dictionary and Text in Serbo-Croatian

DUSKO VITAS — CVETANA KRSTEV

Abstract

The traditionally compiled dictionaries are one of the sources for the construction of the morphological part of electronic dictionary (abbr. e-dictionary) [Courtois, 90], [Gross, 89b]. The problems that arise when the information in the traditional dictionary does not describe the inflectional paradigm in the manner that is required for the construction of the e-dictionary are discussed in the article. These problems are particularly manifested in the construction of e-dictionary for the language with the rich inflection, such as Serbo-Croatian. The advantages as well as the drawbacks of some formal methods that can supply the missing information are outlined. One of them which is based on the interaction between text and e-dictionary is discussed in more details. Applying this method to the construction of the e-dictionary reduces the description of the inflectional paradigm only to the forms that have occurred in text. This method enables a new approach to the selection of the examples accompanying the dictionary entries. In addition, text associated with the corresponding excerpt from the e-dictionary is a suitable base for further applications.

1. Introduction

The main characteristics of Serbo-Croatian (abbr. S-C), one of the south Slavic languages, are its rich inflection and its phonologically based orthography. As a consequence, the phonologically caused alternations as well as the numerous dialect variants are reproduced in the written text. Traditionally composed dictionaries of S-C [SANU], [MS/MH] describe those phenomena. Usually, all the forms necessary to reconstruct the complete inflectional paradigm are given as a part of an entry definition. The structure of some of these definitions is exposed in [Sabo, Vitas]. For instance, in order to construct all the form of the inflectional paradigm for the nouns ending with *-a* in nominative sg. (abbr. ns.) whose base ends with *k*, *g* or *h*, it is also necessary to list the forms for genitive sg. (abbr. gs), dative sg. (abbr. ds), vocative sg. (abbr. vs) and genitive pl. (abbr. gp). For instance, the list *devojke* gs, *devojci* ds, *devojko* vs, *devojaka* gp is associated to the entry *devojka* ns (*girl*).The traditional dictionaries also describe the dialect variants as a separate entries (example: *devojka* = *devojka* = *djevojka* = *divojka*, ns.), althought they are often only graphemic variations.

The morphographemic generator, described in [Vitas, 80], produce all the forms of the inflectional paradigm of an nominal entry by a calculus on form of ns and using a formal morphographemic definition. This morphographemic definition represents the formalized description of the parameters that determine the proprieties of particular forms, such as the type of a declination, the existence of singular or plural forms, the difference in gender between those forms etc. This definition also includes the description of the alternations that occur in the production of particular forms. Thus, for the entry *devojka* mentioned above, the formal definition includes the designation that it is a feminine gender noun, marked as `[+anim]`, with the unmarked case endings of type *e*, as well as the marks that in ds. palatalization occurs (*devojci*) and that in gp. the fleeting *a* is inserted (*devojaka*). The denoted alternations are peculiar to particular entry: for instance, for the noun *slavopojka* (*the song of praise*), which according to traditional classification belongs to the same class as a noun *devojka*, the forms for ds. and gp. are *slavopojki*. Starting from this formal morphographemic definition the classification of inflectional types in S-C that enables the unique class code assignment to every inflectional class is developed [Vitas, 92], based on [Courtois, 89]. For instance *devojka* is in the class N70.04 while *slavopojka* is in the class N72.01. Every nominal inflectional paradigm can be described either by its morphographemic definition or by a regular expression based on a method proposed in [Gross, 89a]. In that way, the inflectional classes are in general defined by the regular expressions. For instance, the following regular expression corresponds to the class N70.04:

$$ka/\text{ns} + ke/(\text{gs+np+ap+vp}) + ci/(\text{ds+ls}) + ku/\text{as} + ko/vs + kom/\text{is} + aka/\text{gp} + kama/(\text{dp+lp+ip})$$

where in the expression of type x/y, x represents the suffix and y the output

value from the finite transductor. That is, the addition of the above expression to the strings it devoj-, *devoj-*, *djevoj-*, *lepoj-*, *troj-* yields all the forms of the inflectional paradigm of the corresponding nouns from the class N70.04. This formal description of the nominal inflectional paradigm represents one component in the construction of the morphological e-dictionary of simple words in S-C, in accordance with methodology developed in LADL [Gross, 89b], [Courtois, 90], [Silberztein, 89], [Courtois, Silberztein, 90]. A sample page from, e-dictionary is given below in the Appendix.

2. The Inconsistency of the Traditional Morphological Description

The possibility to extract automatically the morphographemic information from the morphographemic definition of an entry in the traditional dictionaries of S-C was discussed in [Vitas, Pavlović, Krstev]. The purpose of this procedure is to provide the entry with the necessary input information for the algorithm for nominal generation mentioned above. In particular cases this procedure seems possible. For instance, the description of morphographemic information given with the entry *otac* (*father*) in [MS/MH]: gs(*oca*), vs(*oče*), np(*oci*; *očevi*; *ocevi*) defines the formal parameters in such a way that the reconstruction of the other forms of the inflectional paradigm is possible. However, in the contemporary S-C the plural form *oci* is used only in compound words *gradski oci* (*City Fathers*), *oci nacije* (*Fathers of the Nation*) etc. but never to indicate the plural of the male parent when the form *očevi* is used. The form *ocevi* is obsolete and is not in use any more. Moreover, in the class of nouns ending with *-tac*, all the other examples (for instance, *svetac* (*saint*), *zubatac* (*dentex*), etc.) have the plural forms without the infix *-ev-*. Therefore, the noun *otac* represents in the formal respect the morphological exception despite its regular morphological behavior. Thus, the noun *otac* in the basic meaning has the class code N09.04 while the other nouns ending with *-tac* belong to the class N17.16. Similarly, the noun *trud* is a single entry in [MS/MH] although it actually represents two lemmas: *trud* (*effort*) belonging to the class N13.51 is a noun without the plural forms. On the other hand, np(*trud*) = *trudovi* (*birth throes*), has no singular forms and belongs to the class N15.51. The other important information is also missing from the traditional description. For instance, the value of the parameter `anim` is not given explicitly although the forms of inflectional paradigm depend on it. For instance,

$$\text{as}(\check{c}lan) = \begin{cases} \check{c}lana(member), & \text{if [+anim] (class N07.01)} \\ \check{c}lan(article), & \text{if [-anim] (class N08.01)} \end{cases}$$

$$\text{gs}(drvo) = \begin{cases} drva(wood), & \text{if [-anim] (class N51.01)} \\ drveta(tree), & \text{if [+anim] (class N52.01)} \end{cases}$$

Hence, the reusability of morphological information from the traditional dictionary is not possible and can even lead to the erroneous coding of the inflectional

class, as the example of the noun *otac* shows. Moreover, for some entries the traditional dictionaries do not give any information on morphological behavior for different reasons (intuitive referring to some other entry, as is the case with noun *babadevojka* (*spinster*) where it is expected that the user will intuitively connect *babadevojka* with *devojka* or lack of confirmation of some forms in corpus or imprecise norm in that particular cases, as is the case with the entry as *babađusna* for which the gp is not given in [SANU] while more than one form—*babađusni*, *babađusana*, *babađusna*, etc.—are possible). This shows that the construction of morphographemic definition can not be based only on the existing description and that this process for the purpose of constructing of the e-dictionary requires the careful redaction, as is pointed in [Courtois, 89].

Because of the inadequacy of the traditional description for some entries the complete inflectional paradigm can not be defined. This means that for the same entries some parameters in the previously described morphographemic definition can be left undefined. From the formal point of view, this shortage can be compensated in one of the following ways:

(1) generate the forms with the default values of parameters;
(2) generate only the part of inflectional paradigm for which the forms have been established in the traditional dictionaries;
(3) for the undefined parameters, generate all the possible forms, assigning to them successively all the values from the set of the possible values.

For the entry *babadevojka* these procedures give the following results:

form	result 1	result 2	result 3
ns:	babadevojka	babadevojka	babadevojka
gs:	babadevojke	babadevojke	babadevojke
ds:	babadevojki	?	babadevojki babadevojci
vs:	babadevojko	?	babadevojko babadevojka babadevojke
gp:	babadevojka	?	babadevojka babadevojaka babadevojki

In the result (1) the forms for ds and gp are erroneous, in the result (2) the paradigm is only partially defined while in the result (3) it contains erroneous forms in addition to the correct ones. Thus, all three solutions are only auxiliary. The inference by analogy can also lead to incorrect solutions, as was already shown on the example of the entry *otac*.

3. One Solution

As a possible solution of indicated problems, we suggest a computational lexicographic environment in which the interaction between lexicographer, e-dictionary and e-text is achieved. Namely, assume that the partial morphological e-dictionary of simple words of S-C is available and that the certain e-text that is the part of S-C corpus is being processed. Tagging of e-text can be achieved by means of e-dictionary, as can be seen in the following example. For the character string:

+laž nije samo psihološki, moralni ili lični/ momenat, no je prvenstveno duhovna tvorevina i es-/ tetsko oslobođenje imaginacije, tako da podleže is-/ tim zakonima kojima i druga duhovna stvaranja.

the current version of e-dictionary of simple words gives:

(+)*laž*	,laž .N82.01:	Nfsn-; Nfsa-;
(*nije samo*)		
psihološki,	psihološki.A03.01:	Apαmsn; Apαmsa+; Apαmsv; Apαmpn; Apαmpv;
(,)		
moralni	,moralan.A08.02:	Apβmsv; Apβmpn; Apβmpv; Apαmsn; Apαmsa+; Apαmsv; Apαmpn; Apαmpv;
(ili)		
lični	,ličan.A08.02:	Apβmsv; Apβmpn; Apβmpv; Apαmsn; Apαmsa+; Apαmsv; Apαmpn; Apαmpv;
momenat	,momenat.N04.01:	Nmsn-; Nmsa-;
(, *no je prvenstveno*)		
duhovna	,duhovan.A08.02:	Apβmsg; Apβmsa-; Apβfsn; Apβfsv; Apβnsg; Apβnpn; Apβnpa; Apβnpv; Apαfsn; Apαfsv; Apαnpn; Apαnpa;Apαnpv;
tvorevina	,tvorevina.N70.01:	Nfsn-; Nfpg-;
(*i*)		
estetsko	,estetski.A03.01:	Apαnsn; Apαnsa; Apαnsv;
oslobođenje	,oslobođenje.N60.01:	Nnsn-; Nnsa-; Nnsv-;
imaginacije	,imaginacija.N70.01:	Nfsg-; Nfpn-; Nfpa-; Nfpv-;
(, *tako da podleže*)		
istim	,isti.A03.01:	Apαmsi; Apαmpd; Apαmpl; Apαmpi; Apαfpd; Apαfpl; Apαfpi; Apαnsi; Apαnpd; Apαnpl; Apαnpi;
zakonima	,zakon.N04.01:	Nmpd-; Nmpl-; Nmpi-;

(*kojima i*)		
druga	,drugi.A21.01:	Apαfsn; Apαfsv; Apαnpn; Apαnpa; Apαnpv;
duhovna	,duhovan.A08.02:	Apβmsg; Apβmsa-; Apβfsn; Apβfsv; Apβnsg; Apβnpn; Apβnpa; Apβnpv; Apαfsn; Apαfsv; Apαnpn; Apαnpa; Apαnpv;
stvaranja.	,stvaranje.N60.01:	Nnsg-; Nnpn-; Nnpg-; Nnpa-; Nnpv-;

If the text tagged in this way is the input for the lexicographic analysis in the phase of corpus preparation, then the lexicographer can choose between assigned values of category parameters and, perhaps, correct the inflectional class, so that the achieved result at the end would be:

(+)laž	,laž.N82.01: Nfsn-;
psihološki	,psihološki.A03.01: Apαmsn;
moralni	,moralan.A08.02: Apαmsn;
lični	,ličan.A08.02: Apαmsn;
momenat	,momenat.N04.01: Nmsn-;
duhovna	,duhovan.A08.02: Apβfsn; Apαfsn;
tvorevina	,tvorevina.N70.01: Nfsn-;
estetsko	,estetski.A03.01: Apαnsn;
oslobođenje	,oslobođenje.N60.01: Nnsn-;
imaginacije	,imaginacija.N70.01: Nfsg-;
istim	,isti.A03.01: Apαmpd;
zakonima	,zakon.N04.01: Nmpd-;
druga	,drugi.A21.01: Apαnpn;
duhovna	,duhovan.A08.02: Apβnpn; Apαnpn;
stvaranja	,stvaranje.N60.01: Nnpn-;

or, after substitution in the input string,
+laž (laž. N82.01: Nfsn-) *nije samo psihološki* (psihološki. A03.01: Apαmsn), *moralni* (moralan. A08.02: Apαmsn) *ili lični* (ličan. A08.02: Apαmsn)/ *momenat* (momenat. N04.01: Nmsn-), *no je prvenstveno duhovna* (duhovan. A08.02: Apβfsn; Apαfsn) *tvorevina* (tvorevina. N70.01: nfsn-) *i es-/ tetsko* (estetski. A03.01: Apαnsn) *oslobođenje* (oslobođenje. N60.01: Nnsn-) *imaginacije* (imaginacija. N70.01: Nfsg-), *tako da podleže is-/ tim* (isti. A03.01: Apαmpd) *zakonima* (zakon. N04.01: Nmpd-) *kojima i druga* (drugi. A21.01: Apαnpn) *duhovna* (duhovan. A08.02: Apβnpn; Apαnpn) *stvaranja* (stvaranje. N60.01: Nnpn-).

4. Conclusion

If the text equipped in this way is the part of the corpus, then when processing an entry all actually realized forms of inflectional paradigm can be extracted

which enables the checking of realization of particular forms. It also enables the restriction of some possibilities that morphological system allows but which do never occur. As one result of such interaction between e-text and e-dictionary, the association of text and its dictionary is obtained, that is suitable for further processing.

Acknowledgment

We would like to express our gratitude to Prof. Maurice Gross and Blandine Courtoise, from LADL, for all the support and information that helped our work.

References

Courtois, B.: *Construction du lexique DELAS: Codification et contrôle des entrées lexicales*, LADL, Paris, mai 1989

Courtois, B.; Silberztein, M. (eds.): *Dictionnaires électroniques du français*, Langue française, 87, Larousse, Paris, septembre 1990

Courtois, B.: *Un système de dictionnaires électroniques pour les mots simples du français*, in: [Courtois, B.; Silberztein, M., 90], pp. 11–22

Courtois, B.; Silberztein, M.: *Developement d'outils de navigation dans les grandes documentations*, Rapport technique no. 28, LADL, Paris, novembre 1990

Gross, M. 89a: *The Use of Finite Automata in the Lexical Representation of natural Language*, in: Gross, M.; Perrin, D. (eds.): Electronic Dictionaries and Automata in Computational Linguistics, Lecture Notes in Computer Science, no. 377, Springer-Verlag, Berlin, 1989, pp. 34–50

Gross, M. 89b: *La construction de dictionnaires électroniques*, Annales des télécommunications, 44(1–2), janvier–février 1989, pp. 4–19

MS/MH: *Rečnik srpskohrvatskoga književnog jezika*, vol. 1–6, Matica srpska, Matica Hrvatska, Beograd-Zagreb, 1967

Sabo, O.; Vitas, D.: *Mogućnosti osavremenjivanja izrade rečnika*, Proc. of the 4th Conf. "Computer Processing of Language Data" (Tancig, P. ed.), IJS, Portorož, oktobar 1988, pp. 375–384

SANU: *Rečnik srpskohrvatskog književnog i narodnog jezika*, vol. 1–14 (A–N), Srpska akademija nauka i umetnosti, institut za srpskohrvatski jezik, Beograd, 1959–1990

Silberztein, M.: *Dictionnaires électroniques et reconnaissance lexicale automatique*, Thèse de doctorat en Informatique fondamentale, Université Paris 7, novembre 1989

Vitas, D.: *Generisanje imeničkih oblika u srpskohrvatskom jeziku*, Informatica 3/81, Ljubljana, 1981, 49–55

Vitas, D.; Pavlović, G.; Krstev, C.: *Morphographemic Definitions in Dictionaries of Serbo-Croatian*, [to appear]

Vitas, D.: *Matematički model morfologije srpskohrvatskog jezika (imenska fleksija)*, Matematički fakultet, Univerzitet u Beogradu, Beograd, 1992

Appendix: The Excerpt from the E-dictionary of S-C— Nouns Begining with B

The structure of an entry is:

form, lemma. code of elementary class: codes of forms

Codes of forms are represented by the expression of type $Nabcd$; where a is a mark for gemder, b is a mark for number, c is a mark for case and d is a mark for animateness.

```
          .........
     BABA ,BABO.N42.01:       Nmsg+; Nmsa+;
          ,BABO.N78.01:       Nmpg+;
          ,BABA.N75.01:       Nmsn+; Nmpg+;
          ,BABA.N70.01:       Nfsn+; Nfpg+;
          ,+BABE.N70.75:      Nfpg-;
          ,BABE.N65.02:       Nfsg+;
          ,+BABA.N70.51:      Nfsn-; Nfpg-;
          ,BABA.N70.01:       Nfsn-; Nfpg-;
          ,BABA.N72.01:       Nfsn+; Nfsv+; Nfpg+;
    BABAD ,BABAD.N04.01:      Nmsn-; Nmsa-;
          .........
BABALUSXKU ,BABALUSXKA.N70.01: Nfsa-;
BABALUSXCI ,BABALUSXKA.N70.01: Nfsd-; Nfsl-;
   BABAMA ,BABO.N78.01:       Nmpd+; Nmpl+; Nmpi+;
          ,BABA.N75.01:       Nmpd+; Nmpl+; Nmpi+;
          ,BABA.N70.01:       Nfpd+; Nfpl+; Nfpi+;
          ,+BABE.N70.75:      Nfpd-; Nfpl-; Nfpi-;
          ,+BABA.N70.51:      Nfpd-; Nfpl-; Nfpi-;
          ,BABA.N70.01:       Nfpd-; Nfpl-; Nfpi-;
          ,BABA.N72.01:       Nfpd+; Nfpl+; Nfpi+;
    BABAN ,BABAN.N04.01:      Nmsn-; Nmsa-;
          ,BABAN.N01.01:      Nmsn+;
   BABANA ,BABAN.N04.01:      Nmsg-; Nmpg-;
          ,BABAN.N01.01:      Nmsg+; Nmsa+; Nmpg+;
   BABANE ,BABAN.N04.01:      Nmsv-; Nmpa-;
          ,BABAN.N01.01:      Nmsv+; Nmpa+;
   BABANI ,BABAN.N04.01:      Nmpn-; Nmpv-;
          ,BABAN.N01.01:      Nmpn+; Nmpv+;
 BABANIMA ,BABAN.N04.01:      Nmpd-; Nmpl-; Nmpi-;
          ,BABAN.N01.01:      Nmpd+; Nmpl+; Nmpi+;
  BABANOM ,BABAN.N04.01:      Nmsi-;
          ,BABAN.N01.01:      Nmsi+;
   BABANU ,BABAN.N04.01:      Nmsd-; Nmsl-;
          ,BABAN.N01.01:      Nmsd+; Nmsl+;
   BABARA ,BABARA.N70.01:     Nfsn-; Nfpg-;
```

```
  BABARAMA ,BABARA.N70.01:     Nfpd-; Nfpl-; Nfpi-;
           .........
     BABAC ,BABAC.N17.12:      Nmsn+;
           ,BABAC.N18.12:      Nmsn-; Nmsa-;
    BABACA ,BABAC.N17.12:      Nmpg+;
           ,BABAC.N18.12:      Nmpg-;
    BABACI ,BABAKA.N70.01:     Nfsd-; Nfsl-;
           .........
   BABINXA ,BABINXE.N70.51:    Nfpg-;
  BABINXAK ,BABINXAK.N01.04:   Nmsn-; Nmsa-;
 BABINXAKA ,BABINXAK.N06.02:   Nmpg-;
           ,BABINXAK.N01.04:   Nmsg-; Nmpg-;
           ,BABINXACI.N06.02:  Nmpg-;
 BABINXAKE ,BABINXAK.N06.02:   Nmpa-;
           ,BABINXAK.N01.04:   Nmpa-;
           ,BABINXACI.N06.02:  Nmpa-;
BABINXAKOM ,BABINXAK.N01.04:   Nmsi-;
 BABINXAKU ,BABINXAK.N01.04:   Nmsd-; Nmsl-;
 BABINXAMA ,BABINXE.N70.51:    Nfpd-; Nfpl-; Nfpi-;
           .........
     BAPKA ,BABAK.N04.06:      Nmsg-;
           ,BABAK.N01.06:      Nmsg+; Nmsa+;
     BAPKE ,BABAK.N04.06:      Nmpa-;
           ,BABAK.N01.06:      Nmpa+;
    BAPKOM ,BABAK.N04.06:      Nmsi-;
           ,BABAK.N01.06:      Nmsi+;
     BAPKU ,BABAK.N04.06:      Nmsd -; Nmsl-;
           ,BABAK.N01.06:      Nmsd+; Nmsl+;
     BAPCA ,BABAC.N17.12:      Nmsg+; Nmsa+;
           ,BABAC.N18.12:      Nmsg-;
     BAPCE ,BABAC.N17.12:      Nmpa+;
           ,BABAC.N18.12:      Nmpa-;
    BAPCEM ,BABAC.N17.12:      Nmsi+;
           ,BABAC.N18.12:      Nmsi-;
     BAPCI ,BABAC.N17.12:      Nmpn+; Nmpv+;
           ,BABAK.N04.06:      Nmpn-; Nmpv-;
           ,BABAK.N01.06:      Nmpn+; Nmpv+;
           ,BABAC.N18.12:      Nmpn-; Nmpv-;
   BAPCIMA ,BABAC.N17.12:      Nmpd+; Nmpl+; Nmpi+;
           ,BABAK.N04.06:      Nmpd-; Nmpl-; Nmpi-;
           ,BABAK.N01.06:      Nmpd+; Nmpl+; Nmpi+;
           ,BABAC.N18.12:      Nmpd-; Nmpl-; Nmpi-;
     BAPCU ,BABAC.N17.12:      Nmsd+; Nmsl+;
           ,BABAC.N18.12:      Nmsd-; Nmsl-;
```

Restricted Editing in a Corrected Dictionary Text File

J.J. VAN DER VOORT VAN DER KLEIJ — J.G. KRUYT

Our institute is preparing a computerized version of the *Woordenboek der Nederlandsche Taal*, a historical dictionary equal to the *Oxford English Dictionary*. A machine-readable version of the dictionary is not suited as input for the automatic text encoding. Lexicographical economy requires an intermediate step of text modification. A computer-aided procedure has been developed by which this modification can be performed with minimal risk of corrupting the correct dictionary text file. The modifications are efficiently checked automatically to a major extent.

1 The problem

The Institute for Dutch Lexicology (INL) is preparing a computerized version of the *Woordenboek der Nederlandsche Taal* (WNT; 1882 →), a historical dictionary covering the Dutch language from 1500 up to the 20th century. The Electronic WNT will be a text file encoded for information categories in SGML-format (Bryan 1988), comparable to the electronic *New Oxford English Dictionary* (Kazman 1986).

A correct dictionary text file, conceived as a machine-readable version of the printed text, is not suited as input for the automatic encoding of information categories. It makes no difference wether the file is produced by word processing or by use of Optical Character Recognition (cf. Kruyt & Van der Voort van der Kleij 1992). Lexicographical economy and structural ambiguity raise the need for an intermediate step of text modification. This affects mainly two information categories: sense level and quotation reference. In the printed version, one type of structural ambiguity concerns the number of indentations at the various sense levels, generally being one or two. That is why the computer cannot discriminate between some hierarchically different sense levels that are identical in form (see De Bruin et al. 1991). This ambiguity is solved by altering the number of indentations. Often it is necessary to study the hierarchical structure of the entry before a decision can be made. This is one of the reasons why this revision needs manual intervention.

Lexicographical economy concerns quotation references. In the dictionary, a quotation text is usually followed by the author, the title of the work, page number and the date. The date is commonly enclosed by two square brackets followed by a full stop. In a number of cases the date is left out. For instance when the date is included in the title of the work, or when a quotation text is preceded by a quotation from the same source with the same date. In the latter case, only the first gets the full reference. Another, complex case is shown in the following two quotations in which the second reference only consists of "*Ald.*" ('there'), implying that both author "**HUBNER**", title "*Koer.-tolk*", page "925 *a*" and date "[1732]" are identical to the former reference:

> in 't keurvorstendom Saxen zyn twederly uytschotten ... die by uytschotsdagen beroepen worden, **HUBNER**, *Koer.-tolk* 925 *a* [1732]. In den ruymen uytschot bestaat de ridderschap uyt 60. personen ... en uyt 18. steden als Annaberg, Weissenfels enz., *Ald.*

Dates, being a textual feature that will be used as a marker for "end of quotation", are essential for the automatic separation of quotations and subsequently for the automatic encoding of the various components of the quotation reference. Therefore references without dates need to be supplemented. Automatic insertion of the date is impossible in most cases. Omitted dates and their surrounding square brackets have to be keyed in into the dictionary file at the right locations.

A computer-aided procedure has been developed by which this revision can be performed with minimal risk of corrupting the dictionary text, being an extensively corrected file. Moreover, the inserted modifications are efficiently checked, both during and after the revision process. Final correction is computer-aided as well. The present paper describes the three components of this procedure: revision by means of restricted editing, computational checking of the revised file and final correction.

2 Restricted computer-aided editing

As the revision applies to an extensively corrected dictionary text, it is relevant to protect text fragments that must remain unchanged. Using the extendable VAX-editor EVE in

combination with the programming language VAXTPU (Text Processing Utility), we developed a computer-aided system for restricted editing. EVE has been developed into a dedicated editor by extending it special procedures, which are programs that can be run during the editing session. Some are activated by starting the editor, others by pressing keys. Furthermore, correctness of modifications is checked to some extent already in this stage.

Prior to the proper text modification process, the layout of the dictionary files that are to be revised (each containing about 60 columns), is changed by a program. The lexicographer's text is separated from the quotation block and at the start of each editorial line a copyright sign plus a space are inserted. The quotation block starts at a new line. The beginning of this block is, as in the book, marked by two vertical lines. The output of this program, a file with the extension ".term", is input for the dedicated editor, which only accepts files with that extension to ensure that the input has the right form.

Text modification is performed at a terminal connected with the central computer (VAX 8350 running under the operating system VMS), as the editing program is running on the VAX. Advantages of using our central computer include an automatical back up procedure every night, an easy way of distributing the files to the correctors and the possibility of automatic administration of the whole modification process (the latter two are not yet implemented for this process, but cf. Kruyt & Van der Voort van der Kleij 1992).

When the editor is started, the text file is read from disk and appears on the screen. In the text file, type fonts are represented by graphical codes indicating start and end (for italic for example: QCU*Koer.-tolk*ZCU). The graphical codes for bold, small capital and italic font are replaced by video attributes (reverse video and bold simultaneously, bold, and reverse video, respectively) to clarify the text structure (presently we do not have at our disposal a text display utility like LECTOR used by the OED; cf. Raymond 1990). The graphical codes are reinserted when the editing session is completed. Replacement and reinsertion are not controlled by the corrector but are executed automatically by starting and exiting the editor. Four buffers are visible on the screen in separate windows: the text buffer (18 lines + status line), work buffer (2 lines + status line), message buffer (one line) and the command buffer (one line).

The text window contains the dictionary text. In this buffer, scrolling and searching is possible, but altering the text is inhibited (cf. Kruyt & Van der Voort van der Kleij 1992), because this buffer is made unmodifiable.

Revisions can be inserted via the work window only. When the corrector has positioned the cursor at the correct location in the text window, he presses the date key. The cursor is then placed into the work window. If the cursor would not have been put onto an admitted location, this would not have happened, as the program under that key first checks the symbol at the cursor position and its context in the text window before transferring the cursor to the work window. That symbol must be a full stop followed by a space generally, and the current line must not start with a copyright sign (the marker for lexicographer's text). The left and right context of the full stop has to meet program conditions on text patterns. For defining the conditions, we used a VAX-BASIC program as a tool to analyze a lot of machine-readable fascicles produced for the printing company since 1982. These conditions prevent in most cases that dates are inserted at a wrong location. By similar procedures, the corrected dictionary text is protected to a major extent. In the first quotation in section 1, for example, the full stop preceded by a date ("[1732]") would not be accepted as a place allowing insertion. This prevents changing existing dates. In the following quotations (the ellipses are ours) the full stops after "570" and "*Ald*" are correct locations. The full stops after "*Ned*" and "*Jaerb*", however, are wrong locations but they would nevertheless be accepted having the same form as "*Ald*" (see however section 3 for a solution of this problem).

... zoo zullen zy verbeuren dien Wyn, en daer toe duizend Guldens, mitsgaders hunne neringe,

QCU*Ned. Jaerb.*ZCU 1750, 570. In reguard van de Zeep, uitgeschreven ... zullen dezelve gezwore Zeepwerkers de Billieten ... moeten overbrengen aen den Collecteur, QCU*Ald*ZCU.

When the current full stop is accepted, the corrector may key in the date. If he wants to correct himself, he presses the reset key, the window is emptied and he returns to the text window. Pressing the confirmation key results in a an automatic checking (by the program under the key) of some formal aspects of the typed date, for example the square brackets. If the formal conditions are not met, the corrector receives a message indicating the error. Otherwise the text buffer is made modifiable, the date is transferred into that buffer and the text buffer is made unmodifiable again. In the meantime, the new date is surrounded by number signs ("#") in order to mark dates inserted by the corrector. The work buffer is emptied. The corrector may revise dates he has inserted like he inserts new dates. The number signs help to recognize them. The following example shows the quotations presented above after correct insertion of dates. The two dollar signs, added by the corrector, are an extra marker indicating that the inserted date is based on the year in the title.

... zoo zullen zy verbeuren dien Wyn, en daer toe duizend Guldens, mitsgaders hunne neringe, QCU*Ned. Jaerb.*ZCU 1750, 570 #$$[1750]#. In reguard van de Zeep, uitgeschreven ... zullen dezelve gezwore Zeepwerkers de Billieten ... moeten overbrengen aen den Collecteur, QCU*Ald*ZCU. #$$[1750]#.

The work window is used for comments of the correctors as well. This facilitates an efficient processing afterwards (see section 4). By pressing the comment key, the cursor is located into the work window, and the current line of the text window, preceded by its line number, is copied to the message buffer and to the work window. The corrector may change that line of text and make his comment on the second line of that window. Comments at least concern required alterations of the number of indentations, which are not adapted in the text buffer in this phase (cf. section 3). Comments may furthermore be related to problematic date insertions or detected text irregularities. After pressing the confirmation key, the contents of this window are copied to the message buffer, the work buffer is emptied and the cursor returns to the text window. Use of the reset key has the same result as mentioned before.

Messages to the corrector are shown in the message window. Because this window has one line, only the last message is visible. These messages include proper system messages, for example "Attempt to modify unmodifiable buffer" or "Shutdown of the computer system". Other types of messages concern the acceptability of locations proposed for date insertion (for example "This is not an admitted place for date insertion") and formal correctness of inserted revisions (for example "The square brackets are not correct").

Commands to the system are given in the command prompt window after pressing the command key. For example, the command "Wildcard find" enables special search facilities.

The editing session results in three output files. The first is the original dictionary text file provided with inserted dates. Its extension is changed into ".term_hc" so as to indicate that the revision is completed. The second output file, characterised by the extension ".term_mod", contains all lines of the message buffer and all comments made by the corrector, including his suggestions for indentation alterations. The third output file, with the extension ".term_tim", contains data provided by the computer system about starting time, ending time and duration of the edting session.

In spite of the provisional check on formal aspects during the editing session, a check on lacking or incorrect dates is required. For this purpose, an exhaustive checking program has been written in order to minimize a final human check.

3 Computational check on text revisions

The checking program is written in VAX-BASIC. Input for the program is the revised dictionary text file. Each paragraph, a string commonly existing of lexicographer's text followed by quotations, is successively written into memory. In the quotation block, the program first checks locations where a date might still be lacking, by use of conditions on text patterns and formal features relating to the beginning and end of a quotation. Next, correctness of the inserted dates is checked on the basis of combined conditions on text patterns and contexts. Date checking concerns formal aspects as well as contents of the date. As for the lexicographer's text, some formal aspects are automatically checked, additionally to the correctors comments concerning indentations. Like in the restricted editing process, the checking programs are based on textual patterning, which is in this case more complicated.

The output of the program is a text file reporting the name of each checked file, a list of correctly inserted dates, incorrectly inserted dates, probable locations where dates should have been inserted, and definition lines in which a structural alteration may be required. We will illustrate this with some examples based on our practice (the lines with more than 80 positions are split up).

The following example shows a correct insertion. The line number ("111") is followed by the string "MAT", which indicates that a date is inserted, by a small portion of the context to the left, by the inserted date and by a small portion of the context to the right. The asterisk is indicating the place of insertion.

```
111: MAT U 1918, 1396 QCUaZCU #$$[1918]#. De uniform van de welpen be
                                        *
```

The list of correct insertions also includes insertions at locations not protected. A corrector might have made an insertion at a wrong location, for example at the above mentioned "*Ned*" or "*Jaerb*". We have not found this kind of error until now. However, if present, these errors can be detected very easily, by reading a selective part of the output file.

Fourteen types of potential errors are distinguished, each marked by a bracket plus a type number. This enables the search for special error types. The following example shows the error type "14", a portion of 80 characters of the context to the left, the line number "716", the indication "MAT", a small portion of the immediate context preceding the insertion, the inserted date, and a message about the kind of error ("DAT. FOUT"). In this case, the corrector typed the wrong number "1899", which should have been "1889" corresponding with the date in the title.

```
{14  voor de veiligheid van den alleenloopenden voetganger (QCUop een kermis), Haagsc
716: MAT h Jaarb.ZCU 1889, 29 #$$[1899]#. DAT. FOUT?, <> cijfer |1889|
                                        *
```

Next example shows the function of the context of 80 characters. The message "DAT. ONGELIJK AAN VORIGE" in the output file reports that the inserted date is incorrect because the program has determined that the inserted date "[1938]" is not equal to the preceding date "[Kempen, 1938]". We can verify this by reading the context of 80 characters to the left in the output file.

{10 QKKCORNZKK., QCUBijv.ZCU [Kempen, 1938]. Het uitgewalmde stroo dient om daken t
309: MAT e dekken, QCUAldZCU. #[1938]#. DAT. ONGELIJK AAN VORIGE [Kempen, 1938]

*

The most frequent error type is error type number one, a missed insertion, illustrated by the following example. The indication "DAT?" and the message "NOG TE DATEREN?" means: is here a date lacking? The date "$$[1855]" should have been inserted after "204 *b*".

{1 Goudvernis, dat noch door 't licht, noch door de lucht verschiet, QCUVolksvlijtZ
114: DAT? CU 1855, 204 QCUbZCU. Vroeger kwam het verschieten in de was → NOG TE DATEREN?

The last example shows a definition line in which a structural alteration may be required. "RNR. 1044" indicates the line number. The five spaces in front of "II)" represent one indent. The string "7) Alleen, enkel, slechts.", being a subsense of sense level "II) Bijw.", is to be removed to a new line beginning with one indent.

RNR. 1044 © II) Bijw. __ 7) Alleen, enkel, slechts.

The output of this extensive checking program strongly supports the final correction.

4 Final correction

The final, human checking deals with the output of the checking program and a file containing all corrector's comments. In this phase, we use another extended version of EVE in which three main buffers are employed: a text buffer including the dictionary text, the corrector's comments buffer and a buffer containing the output of the checking program. Two buffers appear simultaneously at the screen. The first always is the text buffer, and the second is either the comment buffer or the buffer with the checking report. The comments and the output of the checking program can be handled quite efficiently by a link between the line numbers in the three buffers. We position the cursor at a numbered comment line or at a numbered report line, and by pressing the link key, the cursor is placed in the text buffer on the corresponding line. This line is highlighted and the comment or report line as well. That way, scrolling and searching is reduced to a minimum. Then we view the comment or report and may modify the dictionary text. The structural indentations are adapted in this phase.

The output is a dictionary text file in which textual adaptations preparing the automatic encoding of information categories have been performed. The new file has the extension ".term_hc_kor".

5 Discussion

A first version of the special editors and of the checking program was developed in 1990-1991. At present, a large number of files have been processed according to the developed procedure. It has proven its value compared to an earlier approach. Before 1990, the text modifications were manually inserted on enlarged copies of the printed dictionary and visually checked. Thereafter, the text was keyed in. The present method has resulted in an improved quality of the product, by use of automatic checking procedures, both during and after the

revision process. Moreover, durations of the revision process are reduced to about one third (two hours instead of six hours per 60 columns). As this process concerns ca. 100.000 columns of dictionary text in total, this reduction of time implies a considerable reduction of man-years required for the project as a whole.

Rather than an automatic check, it would therefore be attractive if the adaptations could be automatically inserted. As for the structural indentations, we already referred to the necessity of a human analysis of the hierarchical structure of the entry (section 1). Automatic insertion of dates requires exact and complete definitions of textual patterns determining the insertion locations. As shown by the example with the strings "*Ned*" and "*Jaerb*", this is not always possible. Additionally, the research into textual definitions is very laborious, as human knowledge on textual patterns is not sufficient. The definitions of the textual patterns used in the procedures described above, have been established by an iterative process of analyzing textual patterns in a lot of machine-readable fascicles. For these reasons, automatic textual adaptation is still a utopia.

On the other hand, textual patterning is very helpful for reducing manual work. Earlier, similar dedicated editors have been developed at our institute, for the preparation of a new version of the official Dutch spelling guide. Part of the revision concerned updating of information fields within the entries. Restricted editing based on textual patterning turned out to be an effective tool in reducing correcting work. This new version, *Herziene woordenlijst van de Nederlandse taal*, was published in 1990 (by SDU, The Hague).

Until now, two projects at our institute have profited from restricted editing based on textual patterning. This method may therefore have a more general interest and be implemented in professional editors (cf. Knowles 1990). This kind of professional editors can be applied not only to dictionary text files, but also to other text files with information categories that are computer definable.

References

Bruin, H.J.B.A. de, J.J. van der Voort van der Kleij, J.G. Kruyt (1991), Algoritmen voor een dictionary entry parser voor het Elektronisch WNT. *INL Working Papers 91-02.*

Bryan, M. (1988), *SGML, An Author's Guide to the Standard Generalized Markup Language.* Addison-Weshley, London.

Kazman, R. (1986), *Structuring the Text of the Oxford English Dictionary through Finite State Transduction.* Doctoral dissertation University of Waterloo, Data Structuring Group CS-86-20.

Knowles, F. (1990), Language and IT: Rivals or Partners? In: *Literary and linguistic computing* 5, 38-44.

Kruyt, J.G & J.J. van der Voort van der Kleij (1992), Towards a computerized historical dictionary of Dutch: from printed dictionary to correct text file. *Proceedings COMPLEX '92.*

Raymond, D.R. (1990), *Lector - An Interactive Formatter for Tagged Text.* University of Waterloo, Department of Computer Science N2L 3G1.

List of Participants

SUE ATKINS
Oxford University Press
Walton Strret
Oxford, England, OX2 6DP

VLADIMIR BENKO
Computational Linguistics Laboratory Comenius University
Moskovská 3
Bratislava, CSFR, CS-813 34

CHRISTOPH BLÅSI
Lehrstahl für Computerlinguistik, Fakültat für Linguistik und Literaturwissenschaft, Universitat Bielefeld
Posfach 100 131
Bielefeld-1, Deutschland, D-4800

ANNA BRAASCH
Center for Spragteknologi Kóbenhavos Universitet
Njalsgade 80
Copenhagen S, Danmark, DK-2300

DANIEL BRESSON
Université de Pronvence, Dt d'allemand
29 av R. Schuman
Aix en Provence, France, F 13621

STEPHEN BULLON
Cobuild
Westmere 50 Edgebaston Park Road
Birmingham, United Kingdom, B15 2RX

DAVID CLEMENCEAU
Laboratoire d'Automatique Documentaire et Linguistique, Universite PARIS 7
2, place Jussieu
Paris, France, 75251

MARIE-HELENE CORREARD
Oxford-University Press
Walton Street
Oxford, United Kingdom, OX2 6DP

BLANDINE COURTOIS
CNRS LADL Ubiversit; Paris 7
2 place Jussieu Tour Centrale 9 etage
Paris, France, 76221

PAT COWAN
Harper Collins Publishers
P.O. Box
Glasgow, United Kingdom, G4 0NB

GALINA DIANOVA
Moscow University Faculty of Modern Languages

Moscow, Russia, B-234

ISTVÁN DIENES
Centrum Statistical Office
P.O.B. 51.
Budapest, Magyarország, 1525

DANIELLE DUJARDIN
Laboratoire de Génie Informatique (IMAG)
IMAG-CAMPUS BP 53 X
Grenoble, France, 38041

STEFANO FEDERICI
ILC-CNR
Via della Faggiola 32.
Pisa, Italy, 56100

THIERRY FONTENELLE
University of Liege, English Department
3, Place Cockerill
Liege, Belgium, B-4000

MME AGGELIKI FOTOPOULOU
Université Paris 7 L.A.D.L.
2 Place Jussieu
Paris, France, 75005

KATHARINA GREWE
Sprachwissenshaftliches Institut
Postfach 10 21 48
Bochum 1, Germany, D-4630

MAURICE GROSS
L.A.D.L. Universite Paris 7
2, Place Jussieu
Paris, France, 75221

VALERIE GRUNDY
Oxford-University Press
Walton Street
Oxford, United Kingdom, OX26DP

ULRICH HEID
Institut für Machinelle Sprachverarbeitung - Computerlinguistil
Azenbergtsratss 12
Stuttgart, Germany, DW-7000

ANNAMÁRIA KABÁN
Kolozsvári Nyelv- és Irodalomtudományi Intézet
Str. Umirii Nk. 3. ap. 27
Cluj, Románia, 3400

ILONA KASSAI
Research Institute for Linguistics of the Hungarian Academy of Sciences
Budapest, P.O.Box 19.
Hungary, H-1250

FERENC KIEFER
Research Institute for Linguistics of the Hungarian Academy of Sciences
Budapest, P.O.Box 19.
Hungary, H-1250

GÁBOR KISS
Research Institute for Linguistics of the Hungarian Academy of Sciences
Budapest, P.O.Box 19.
Hungary, H-1250

LAJOS KISS
Research Institute for Linguistics of the Hungarian Academy of Sciences
Budapest, P.O.Box 19.
Hungary, H-1250

LÁSZÓ KISS
Akadémiai Kiadó
Prielle Kornélia u. 19-35.
Budapest, Hungary, 1117

GABRIELLA KLARSFELD
LADL Universite Paris 7
2 Place Jussieu Tour Centrale
Paris, France, 75005

HEINZ DETLEV KOCH
Chair for Computational Linguistics,
University of Heidelberg
Karlstrase 2
Heidelberg, Germany, D-6900

IRENE KOWARSKI
Laboratoire de Génie Informatique
IMAG-CAMPUS BP 53 X
Grenoble, France, 38041

JAN KRALIK
Czech Language Institute
Letenska 4
Praha, CSFR, 118 51

JOHANNA GEERTRUIDA KRUYT
Institute for Dutch Lexicology
P.O. Box 9515
Leiden, The Netherlands, 2300 RA

M. JACQUES LABELLE
D. de Linguistique UQAM
CP 8888 Suc.A
Montréal P.Q., Canada, H3C 3P8

M. ERIC LAPORTE
LADL
Ceril, 25 cours Blaise-Pascal
Évry, France, 91000

M. CHRISTIAN LECLERE
Université Paris 7, L.A.D.L.
2 Place Jussieu (T.C. 9)
Paris, France, 75005

TAMÁS MAGAY
Akadémiai Kiadó
Napos u. 5/b.
Budapest, Hungary, 1125

ELISABETTA MARINAI
ACQUILEX Project, ILC-CNR
Via della Faggiola 32.
Pisa, Italy, 56100

WILLY MARTIN
Institute of Lexicology
Faculty of Arts Free University Amsterdam
De Boelelaan 1105
Amsterdam, The Netherlands, HV 1081

MONICA MONACHINI
ILC-CNR
Via della Faggiola 32.
Pisa, Italy, 56100

JEE-SUN NAM
Universite Paris 7 L.A.D.L.
2 Place Jussieau, Cedex 05
Paris, France, 75251

OLE NORLING-CHRISTENSEN
University of Copenhagen, The Danish Dictionary
Njalsgade 80
Copenhagen S, Danmark, DK-2300

CHRISTIAN-EMIL ORE
Dept. of Lingusitics, University of Oslo
P. O. Box 1102 Blindern
Oslo, Norway, N-0317

JÚLIA PAJZS
Research Institute for Linguistics of the
Hungarian Academy of Sciences
Budapest, P.O.Box 19.
Hungary, H-1250

KAREL PALA
Filozofická fakulta UJEP
A. Nováka 1
Brno, Csehszlovákia, 60200

FERENC PAPP
Research Institute for Linguistics of the
Hungarian Academy of Sciences
Budapest, P.O.Box 19.
Hungary, H-1250

CAROL PETERS
Istituto di Elaborazione della Informazione, CNR
Via della faggiola 32.
Pisa, Italy, 56100

EUGENIO PICCHI
Instituto di Linguistica Computazionale CNR
Via della Faggiola 32
Pisa, Italy, 56100

MIREILLE PIOT
LADL - CNRS Univeriste Paris 8
30, rue Chapon
Paris, France, 75003

VITO PIRELLI
ILC-CNR
Via della Faggiola 32.
Pisa, Italy, 56100

GÁBOR PRÓSZÉKY
MORPHOLOGIC

Budapest, Fő u. 56-58. I/3.
Hungary, H-1011

EMMANUEL ROCHE
LADL Universite Paris 7
2 place Jussieu, Tour centrale
Paris, France, 75221

ADRIANA ROVENTINI
ILC-CNR
Via della Faggiola 32.
Pisa, Italy, I-56100

M. MORRIS SALKOFF
Universite Paris 7 - L.A.D.L.
2 Place Jussieu, Tour centrale
Paris 5, France, 75221

LÁSZLÓ TIHANYI
Research Institute for Linguistics of the Hungarian Academy of Sciences
Budapest, P.O.Box 19.
Hungary, H-1250

FRANK W.M. TOMPA
University of Waterloo,
Centre for the New OED and Text Research

Waterloo, Ontario, Canada
N2L 3G1

JOHN VAN DER VOORT VAN DER KLEIJ
Institute for Dutch Lexicology INL
P.O. Box 9515 .
Leiden, The Netherlands, 2300 RA

JEAN VILLAIN

1 bis rue de L'Ancienne Mairie
Boulogne-Billancourt, France, 92100

ILDIKÓ VILLÓ
Research Institute for Linguistics of the Hungarian Academy of Sciences
Budapest, P.O.Box 19.
Hungary, H-1250

DUSKO VITAS
Computer Laboratory Faculty of Science
Studentski trg 16
Beograd, Yugoslavia, 11000

HENNIE VAN DER VLIET
Vrije Universiteit Amsterdam, Faculteit der Letteren
De Boolelaan 1105
Amsterdam, The Netherlands,

JEAN-MICHEL WALLE
IRIN-LIANA University of Nantes
3, rue du Marechal Joffre, Cedex 01
Nantes, France, 44041

BOJE WANGENSTEEN
Department of Scandinavian Studies and Comparative Literature Section for Lexicography, University of Oslo
P. O. Box 1001 Blindern
Oslo, Norway, N-0315

DAGFINN WORREN
Department of Scandinavian Studies and Comparative Literature Section for Lexicography University of Oslo
P.O. Box 1001 Blindern, Oslo, Norway, N-0315

JUDIT ZIGÁNY
Akadémiai Kiadó
Prielle Kornélia u. 19-35.
Budapest, Hungary, 1117

Hozott anyagról sokszorosítva
9220607 **AKAPRINT** Nyomdaipari Kft. Budapest. F. v.: dr. Héczey Lászlóné